Statistical Methods for the Analysis of Error-Prone Electronic Health Records: Impact of Source Data Verification, Time Discretized Multiple Imputation, and Variance Estimation with Incompatible Imputation and Analysis Models

By

Mark Joseph Giganti

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Biostatistics

August 31, 2018

Nashville, Tennessee

Approved:

Jonathan Schildcrout, Ph.D.

Bryan Shepherd, Ph.D.

Qingxia (Cindy) Chen, Ph.D.

Peter Rebeiro, Ph.D.

ACKNOWLEDGEMENTS

The completion of this dissertation is the end result of years of support from many people. I am grateful for the patient mentorship of my advisor, Bryan Shepherd. His support and guidance has been immense and it has been my privilege to learn from him. I am thankful for my other committee members - Jonathan Schildcrout, Cindy Chen, and Peter Rebeiro - for their valuable insight and contributions to my research. I am thankful to Cathy McGowan and my colleagues at CCASAnet for their support. Our collaborative work inspired and motivated most of this dissertation and it has been a pleasure to work with them. I am fortunate to have the opportunity to learn from many different Vanderbilt Biostatistics faculty and I appreciate their willingness to make time for me and my fellow classmates. I would be remiss if I did not acknowledge those fellow Biostatistics graduate students, especially Allison Hainline, Lucy D'Agostino McGowan, and each classmate in my cohort (David Schluter, Christina Tripp Saunders, Derek Smith, and Minchun Zhou), that helped make graduate school a more pleasant experience. I would like to thank Blake Stepp for the inspiration resulting from his unconscious three. Finally, I would like to thank my wife, Noelle, for her unwavering support of me and my quest to finish this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADE AIDS-defining event

ART Antiretroviral treatment

CCASAnet Caribbean, Central, and South America network for HIV epidemiology

CD4 CD4 cell count

CI Confidence interval

EHR Electronic health records

HR Hazard Ratio

HST Hughes, Sterne, and Tillin

IQR Interquartile range

MI Multiple imputation

MSE Mean square error

RR Rubin's Rules

RW Robins and Wang

SDV Source data verification

TDMI Time-discretized modeling and imputation

VCCC Vanderbilt Comprehensive Care Clinic

VL Viral load, number of copies of HIV RNA in a millilitre of blood

CHAPTER 1

INTRODUCTION

Data from electronic health records (EHRs) provide an invaluable resource for medical research. A pressing concern, however, regarding these observational data sources is their susceptibility to errors due to inaccurate, incomplete, or discordant entries (Weiskopf and Weng, 2013). These errors, typically correlated across multiple variables and difficult to identify, can translate into incorrect epidemiological inferences. One strategy to assess EHR data quality is source data verification (SDV). This data audit procedure compares the research study data to the original source document for a subset of records and documents discrepancies. Given the resource-intensiveness of SDVs, it is imperative to be able to justify their continued implementation. Unfortunately, common practices for doing so are unsatisfactory. In this dissertation, we propose a robust methodological framework to better assess the impact of SDV on epidemiological inference and incorporate validated data into subsequent statistical analyses.

The quality of study datasets is often evaluated based on the frequency of errors they contain. Previous studies conducted in clinical trial settings (Mitchel et al. (2011); Smith et al. (2012)) have shown that higher error rates do not always imply incorrect epidemiological inferences. In Chapter 2, we propose a framework for assessing the impact of audits on study results and illustrate its implementation using a data audit from an international HIV setting as a practical example. First, we quantify error rates for key demographic and clinical variables among audited patients. Then, we demonstrate how errors can affect analyses by performing the same statistical analyses using pre-audit data and audited data for the subset of audited patients. Acknowledging that data audits often identify systematic errors in data collection and entry that lead to targeted validation of high-error-rate variables for all records, we also assess the impact of data audits by performing the same statistical analyses using pre-audit data and post-audit data for all records.

Assuming the discrepancies in the originally collected data are substantial enough to impact epidemiological inferences, it is imperative that investigators address these errors. Current existing strategies when non-trivial errors are revealed include either removing the data in question or re-entering all data. These practices seem impractical. In Chapter 3, we propose a method to obtain unbiased and efficient estimates in time-to-event analyses while incorporating both the original error-prone data for

all subjects and the audited data for the subsample of subjects. The data setup (an error-prone measurement for all records and a "gold standard" measurement for a subset of records) resembles a measurement error and missing data problem. Previous work by Cole et al. (2006), Shepherd et al. (2012), and Edwards et al. (2013) have demonstrated that multiple imputation procedures can be implemented to address similar measurement error scenarios. Although relevant, none of this work or any of the measurement error literature considers the situation where there are errors, likely correlated, in predictors, censored failure times, event indicators, and inclusion criteria, which is what we see in our setting. We propose a time-discretized modeling and imputation (TDMI) approach that uses discrete time models built in a validation sample to multiply impute covariate and outcome values in the remaining unvalidated records.

Having implemented a multiple imputation procedure to obtain unbiased estimates, the final task is to calculate the corresponding imputation variance estimate and report 95% confidence intervals. One possible choice for calculating standard errors is the multiple imputation variance estimator proposed by Little and Rubin (2014). However, this variance estimator has been shown to be biased when the imputation model is misspecified or if there is incompatibility between the imputation model and the analysis model. With our TDMI approach, there is incompatibility because the imputation model is based on time-discretized data with multiple observations for each subject while the analysis model is based on undiscretized data with a single observation for each subject. Additionally, subjects that are excluded in the analysis model for not meeting inclusion criteria remain in the imputation model. An alternative choice for calculating standard errors that allows for incompatibility was proposed by Robins and Wang Robins and Wang (2000). Unfortunately, this approach is complex and there is no publicly available code to facilitate its implementation. In Chapter 4, we provide a tutorial for calculating this imputation variance estimator using multiple examples and by providing comprehensive R code. Examples are chosen to illustrate implementation across multiple different imputation models and analysis models.

CHAPTER 2

THE IMPACT OF DATA QUALITY AND SOURCE DATA VERIFICATION ON EPIDEMIOLOGIC INFERENCE: A PRACTICAL APPLICATION USING HIV OBSERVATIONAL DATA

2.1 Introduction

Source document verification (SDV) is a strategy for HIV research data quality assessment. Typically, SDV involves the partial (or complete) comparison of research study data against original source documents, such as study case report forms, patient clinical charts, laboratory reports, or electronic health records. This practice of data auditing allows investigators to verify data accuracy, identify systematic issues with research data collection, and calibrate their confidence for making inferences based on study findings.

Concerns regarding data quality are magnified for studies using routinely collected observational data from international HIV cohorts. Given that many HIV observational datasets were originally created for clinical or administrative purposes (e.g., electronic health records), data are susceptible to errors with respect to completeness, correctness, concordance, plausibility, and timeliness (Weiskopf and Weng, 2013). Studies assessing HIV observational data quality in multiple international settings have identified data discrepancies and high error rates in key variables (Kiragga et al. (2011); Nicol et al. (2016); Muthee et al. (2018); Puttkammer et al. (2016)). In our own research, we previously audited a subsample of records from several clinical care sites in a multiregional database of pooled HIV treatment data and found both systematic inconsistencies in how data were entered as well as errors that were not flagged by computer-generated error reports (Duda et al., 2012).

Because SDV is resource-intensive - identifying relevant patient records, locating the original source documents, traveling to local sites (for external auditors), comparing source documents to the current research dataset, and recording discrepancies - it is becoming increasingly important to justify the expenses that accompany this task. Many data audits assess data quality according to whether the error rate is above or below an arbitrary 5% threshold (Houston et al., 2015). However, as shown in clinical trial settings (Mitchel et al. (2011); Smith et al. (2012)), high error rates do not necessarily translate into incorrect statistical inferences. In addition to quantifying error rates, the importance of the SDV process should be assessed by investigating potential improvements in data quality at the research network over time and the

overall impact of errors on analyses and corresponding conclusions.

In this study, we assess the impact of SDV audits on observational study results within a multi-cohort, international collaboration. External auditors traveled to sites and conducted SDV for all key HIV study variables on a randomly selected subset of patient records. After the audits, local sites received a report detailing audit findings and recommendations, which in certain cases included requests to re-enter error-prone variables for all patient records in their database. In this manuscript, we describe error rates uncovered by the audits and their impact on study results among the subset of patient records that were audited. We also describe the impact of audits by performing identical analyses using data from the entire cohort, just before the audit and then two years after the audit, to investigate changes made to databases and their impact on key study findings.

2.2 Methods

2.2.1 Cohort description

The Caribbean, Central, and South America network for HIV epidemiology, also known as CCASAnet, is a consortium of clinics from seven Latin American countries that collects and shares HIV care and treatment data across clinical care sites to characterize the HIV epidemic in the region. Nine CCASAnet cohorts participated in this study and contributed data: Hospital Fernández and Centro Médico Huésped in Buenos Aires, Argentina (HF/CMH-Argentina); Instituto Nacional de Infectologia Evandro Chagas in Rio de Janeiro, Brazil (INI-Brazil); Fundación Arriarán in Santiago, Chile (FA-Chile); Le Groupe Haïtien d'Etude du Sarcome de Kaposi et des Infections Opportunistes in Port-au-Prince, Haiti (GHESKIO-Haiti); Instituto Hondureño de Seguridad Social and Hospital Escuela Universitario in Tegucigalpa, Honduras (IHSS/HE-Honduras); El Instituto Nacional de Ciencias Médicas y Nutrición Salvador Zubirán in Mexico City, Mexico (INCMNSZ-Mexico); and Instituto de Medicina Tropical Alexander von Humboldt in Lima, Peru (IMTAvH-Peru). Study data were collected and harmonized at the CCASAnet Data Coordinating Center at Vanderbilt University (CDCC-VU). The CCASAnet cohort has been described elsewhere (McGowan et al., 2007): additional information is at https://www.ccasanet.org.

2.2.2 Data auditing

In 2013-14, on-site audits of submitted data were conducted through a joint effort between data auditors from the CDCC-VU and investigators at all nine participating

sites. For each site, approximately 30 patient records were randomly selected by the data manager at the CDCC-VU to be audited. In advance of the arrival of the data auditors, each site was asked to locate the source documents pertaining to these patients. Source documents available at the sites included paper-based patient charts from the HIV clinic, general hospital charts, laboratory result forms (both paper and electronic), and electronic medical record systems.

An audit team from the CDCC-VU, consisting of at least one clinician and one informaticist, traveled to each of the nine sites. The audit team had a multi-page paper audit form, prepared and checked by the CDCC-VU data manager, displaying all submitted research data for each patient record selected. The auditors were given access to the corresponding source documents for these patients by the collaborators at each site. Over the course of 2-3 days, the data audit team compared values in the research database with the source documents. Each entry was labeled with an audit code (A1-A5) adapted from standardized audit codes defined by the European Organization for the Research and Treatment of Cancer Vantongelen et al. (1989): value matches source document (A1), discrepancy between database and source document (A2 if minor discrepancy, A3 if major), new value in source document not previously entered in database (A4), and value could not be verified in source document (A5). A discrepancy was considered major if the auditors considered the difference between the source and database values to be clinically meaningful; minor discrepancies referred to less significant errors (e.g., date values within a month). New information identified from the source document (A2, A3, or A4) was noted on the paper audit form. All audit findings were later transcribed from the paper audit forms to a study database by the CDCC-VU.

2.2.3 Available data

As part of CCASAnet research collaboration, each site regularly submitted to the CDCC-VU a dataset containing records for all past and present enrolled patients to the CDCC-VU. Prior to the audit, the most recent submission from each site was archived. These site-specific datasets containing records for all enrolled patients were aggregated to generate a *pre-audit dataset*. Approximately two years after the audit (October 2016), the CDCC-VU again archived the most recent submission from each site and aggregated records for all patients from each site to generate a *post-audit dataset*. This time frame encompassed 1-2 scheduled data submission cycles for each site, thereby allowing enough time for audit recommendations potentially to

be incorporated in the new dataset. As for this study, this post-audit dataset was intended to provide a snapshot of changes resulting from the audit. That being said, we note two key clarifications: (1) any data points after the last pre-audit date for a given patient were removed from the post-audit dataset so that pre- and post-audit datasets covered the same time period; and (2) patients records not present in the pre-audit dataset but present in the post-audit dataset were removed, even if they were enrolled prior to the pre-audit freeze date. Lastly, an *audited dataset* was generated for the subset of records that were audited. This dataset contained patient records according to the source document verification findings.

All three datasets contained the same 19 unique variables (as defined and standardized in the CCASAnet data transfer protocol) that were routinely submitted by CCASAnet sites. We refer to these variables as primary variables. For this study, we also generated 14 additional variables that are relevant for our statistical analyses. These derived variables were typically calculated using one or more of the primary variables (e.g., the CD4 cell count at time of antiretroviral treatment [ART] initiation). A complete list of all variables considered for this study is included in Appendix A (Table 2.1).

2.2.4 Statistical analysis

For this study, we defined a data discrepancy as an instance where recorded values were different (A2, A3) or a value was missing in one of the two datasets (A4). When comparing the audited dataset with the pre-audit dataset, we also counted instances where a value could not be confirmed (A5) as a discrepancy. We calculated discrepancy (error) rates for both the originally collected and derived variables used in analyses between (1) the pre-audit and audited datasets in the subset of records that were audited, and (2) the entire pre-audit and post-audit datasets.

To assess the impact of errors identified during a data audit on a typical statistical analysis, we replicated the same statistical analyses in all datasets. We estimated the overall and country-specific (when data were available) cumulative incidences for both the time from ART initiation to death and the time from ART initiation to first ADE (accounting for mortality as a competing risk). A multivariable Cox regression model was fit for each dataset to estimate the hazard ratios (HRs) for predictors of death and ADEs after ART initiation. All models were adjusted for the following covariates: age, sex, probable route of HIV transmission, clinical history of AIDS-defining event, CD4 cell count, initial treatment regimen, and calendar year. All Cox models were

stratified by site to allow the underlying hazard function to differ for each site (Giganti et al., 2015) and used restricted cubic splines (Shepherd and Rebeiro, 2017) with four knots for continuous variables to relax assumptions regarding linearity. Patients were excluded from the study if they were not adults (< 18 years) or never initiated ART. Two countries had multiple sites (Argentina and Honduras); for this analysis, we combined sites within a country into a single site.

All statistical analyses were performed using R Statistical Software (http://www.R-project.org); the corresponding statistical code is available at the following website: http://biostat.mc.vanderbilt.edu/ArchivedAnalyses. Institutional review board approval was obtained from each site and from the Vanderbilt University Institutional Review Board.

2.3 Results

A total of 316 patient records from nine CCASAnet sites were selected to be audited using stratified random sampling by site. The CDCC-VU data auditors reviewed 250 of the selected records during the audit visits. The remaining 66 records were not audited, mainly due to insufficient time during the audit visits or unavailable source documents (including lost, accidentally destroyed, permanently archived charts, and charts currently in use for patient care). The number of audited records varied by site, ranging from 12 at CMH-Argentina to 31 at INI-Brazil. Summaries of audit frequency by site are provided in Appendix A (Table 2.2).

2.3.1 Audited records: pre-audit versus audit data

The pre-audit dataset for these 250 patients contained 19,296 values across 21 variables; 14,496 (75%) were audited. Due to both time constraints and incomplete source documents, some records were only partially audited. Overall, the discrepancy rate across all audited variables was 17.1%. Most discrepancies were due to missing values (43%); the remaining discrepancies were due to incorrect data entries (34%) and data that could not be confirmed by available records (23%). The discrepancy rate differed for each variable, ranging from 1% for sex to 50% for the date of clinical endpoints (Figure 2.1a). Among variables typically collected at time of enrollment, error rates were lower for sex (1%) and birth date (4%), compared to the probable mode of transmission (14%). Only 5% of patients had an incorrect death status when compared to clinic source documentation, yet approximately 25% of all audited death dates had a discrepancy. Date variables had a higher than average percent of

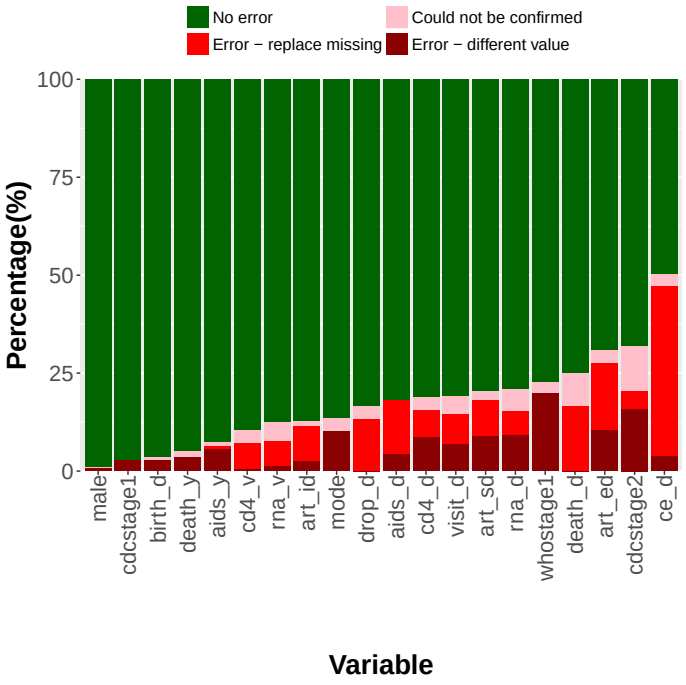
discrepancies, including 31% for ART regimen end dates and 50% for clinical event dates. Error rates for all audited variables are included in Appendix A (Table 2.3 and Figure 2.6).

Of the 250 audited patients, 228 originally met inclusion criteria for analyses (adult patients who initiated ART) in the pre-audit dataset and 232 in the audited dataset; 227 met inclusion criteria in both datasets. Of the five patients excluded from the pre-audit dataset only, four had discrepancies in ART data (2 with missing entries, 1 with an incorrect regimen, and 1 with an incorrect date); the last patient was missing follow-up data. For the single patient excluded in the audited dataset only, a revised birthdate revealed the patient was under 18 at time of ART initiation. For patient records present in both datasets, discrepancy rates for derived variables ranged from 3% to 36% (Figure 2.1b). Variables with the highest error rates corresponded to derived time-to-event variables such as time from ART initiation to first ADE (36%) and follow-up time (32%).

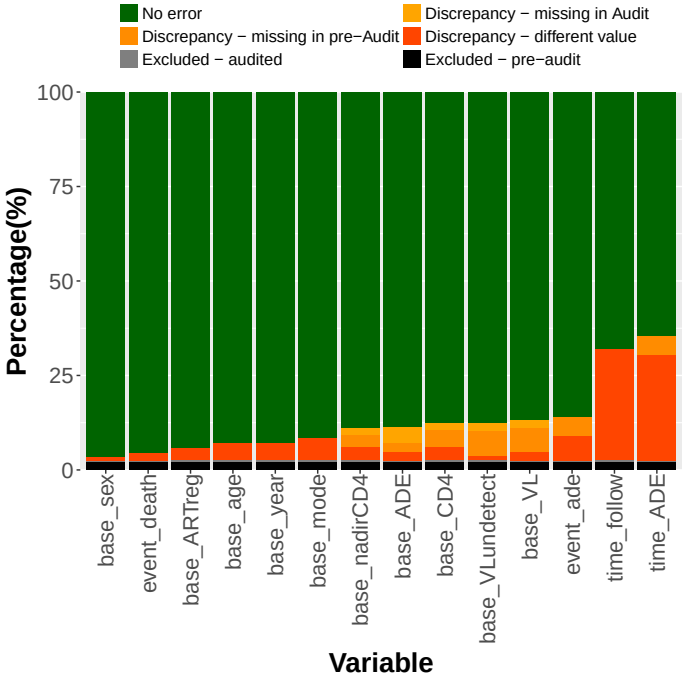
Unadjusted estimates of mortality over time (Figure 2.2a) were similar between audited patients in the pre-audit dataset and the audited dataset. The three-year estimated mortality was 4.2% (1.5%-6.8%) in the pre-audit dataset and 4.1% (1.4%-6.7%) in the audited dataset. Meanwhile, the overall estimated probability of ADE over time was higher in the audit dataset (Figure 2.2b). The estimated percentage of patients with an ADE at three years was 12.9% (7.8%-17.6%) in the pre-audit dataset and 17.5% (11.9%-22.7%) in the audited dataset. Due to the small number of events among the subset of audited records, there was overlap in the confidence intervals for all hazard ratios (Appendix A, Figure 2.7).

2.3.2 Full dataset: pre-audit versus post-audit data

The full pre-audit database included 19,331 adult patients from nine CCASAnet sites. The post-audit dataset, which includes sites' data revisions in response to the audit findings, includes 22,146 eligible adult patients from the same time period as the pre-audit dataset (e.g., with enrollment dates prior to the site-specific freeze dates for the pre-audit dataset.) The post-audit revisions produced a dataset with 18,999 patients from the pre-audit dataset plus 3147 newly added patients. Some patients (n=332) previously included in the pre-audit dataset were not present in the updated dataset. The majority of these patients (n=178) were from a single site that had recently reentered their entire database; duplicate records or instances where the original paper forms could no longer be located were removed. For the

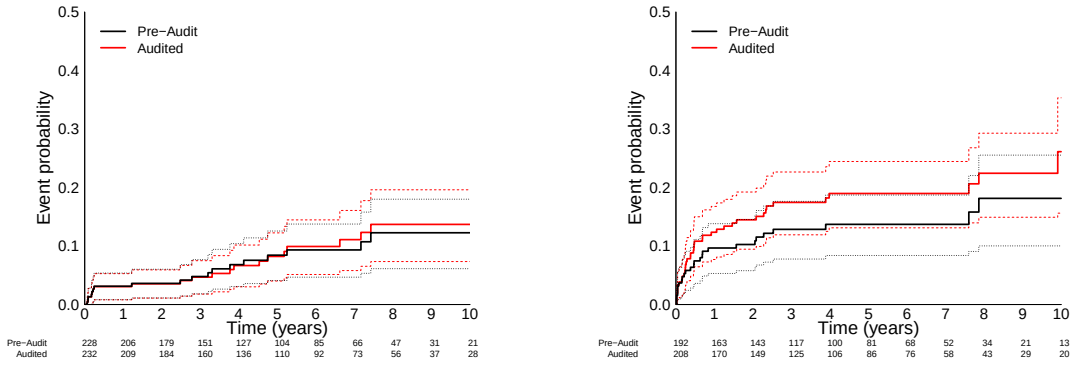


(a) Primary



(b) Derived

Figure 2.1: Relative frequency of discrepancies between pre-audit and audited values for originally collected variables and those derived for analysis.



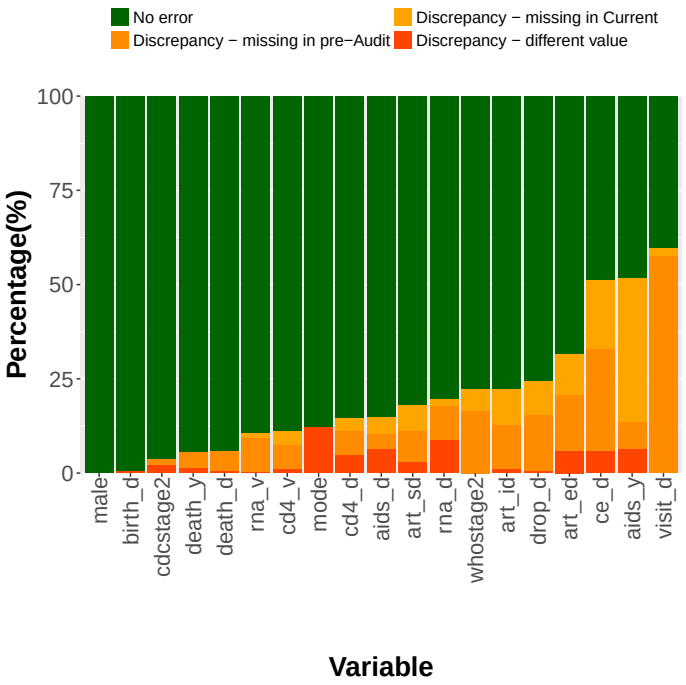
(a) Mortality

(b) ADE

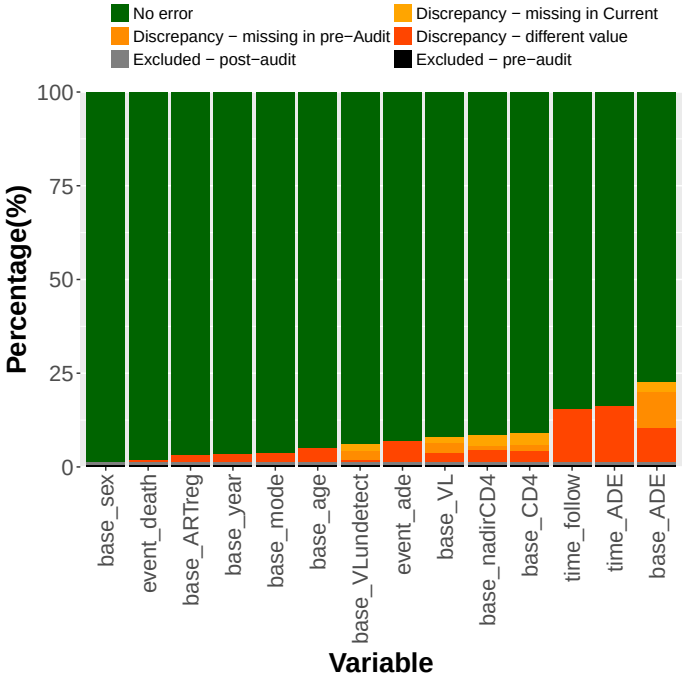
Figure 2.2: Unadjusted time to mortality and AIDS-defining event using pre-audit and audited data, among the subset of patient records that were audited.

18,999 patients in both databases, 1,727,710 unique values were recorded across 19 variables in either the pre-audit or post-audit dataset. Of these, 1,135,693 (66%) were identical in both datasets. The plurality (n=478,600; 81%) of the discrepancies between the two datasets was due to missing values in the pre-audit dataset that were subsequently included in the post-audit dataset. Missing values in the post-audit dataset that existed in the pre-audit dataset explain 12% of discrepancies (n=71,709) and conflicting values accounted for the remaining 7% (n=41,708). The variables with the highest proportion of entries with discrepancies were the date of diagnosis of a clinical endpoint (51%), the occurrence of an AIDS-defining event at baseline (52%) and the date of clinic visit (60%) (Figure 2.3a).

A total of 15,229 patients met inclusion criteria (adult patients who initiated ART) in both the pre-audit and post-audit datasets. Among the remaining 3,770 patients with records in both datasets, an additional 212 were classified as adult ART initiators in only one dataset (124 in the pre-audit dataset only and 88 in the post-audit dataset only). Discrepancy rates for derived variables among 15,441 patients that met inclusion criteria for at least one dataset ranged from 2% for sex to 23% for clinical AIDS status at baseline (Figure 2.3b). Compared to the error rates from the audited subset of records alone, most variables had a lower relative frequency of discrepancies in the post-audit dataset. The key exception was occurrence of an AIDS-defining event at baseline (23% vs. 12%). The unadjusted estimates of mortality over time (Figure 2.4a) were similar between audited patients in the pre-audit dataset and the audited dataset. The three-year estimated mortality was 6.9%



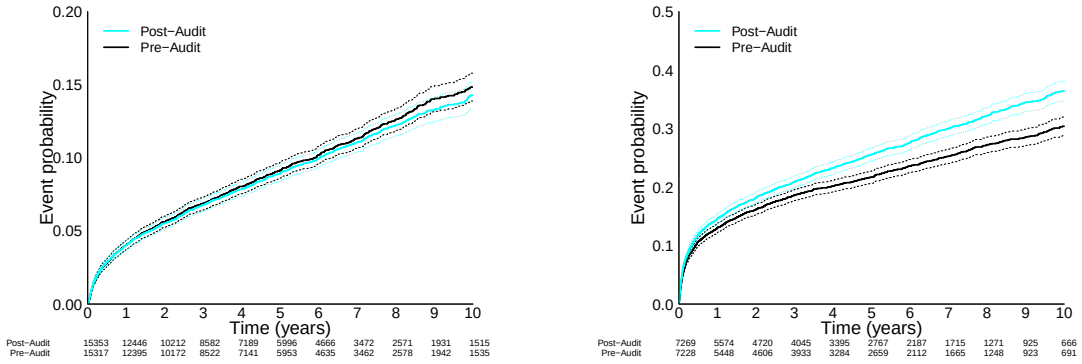
(a) Primary



(b) Derived

Figure 2.3: Relative frequency of discrepancies between pre-audit and post-audit values for originally collected variables and those derived for analysis using patient-specific freeze dates.

(6.4%-7.3%) in the pre-audit dataset and 6.8% (6.3%-7.2%) in the post-audit dataset. The overall estimated probability of ADE over time was higher in the post-audit dataset (Figure 2.4b). The estimated percentage of patients with an ADE at three years was 18.6% (17.8%-19.6%) in the pre-audit dataset and 20.9% (19.9% - 21.9%) in the post-audit dataset. Changes in ADE rates (Appendix A, Figure 2.8) and mortality rates (Appendix A, Figure 2.9) varied by site. Four of the seven sites had similar mortality estimates; two had lower estimates and one had higher estimates using the post-audit dataset. ADE estimates varied for all five regions with available data; estimates were higher for three sites and lower for two sites. ADE data were not available at both time points for two sites, which were therefore excluded.

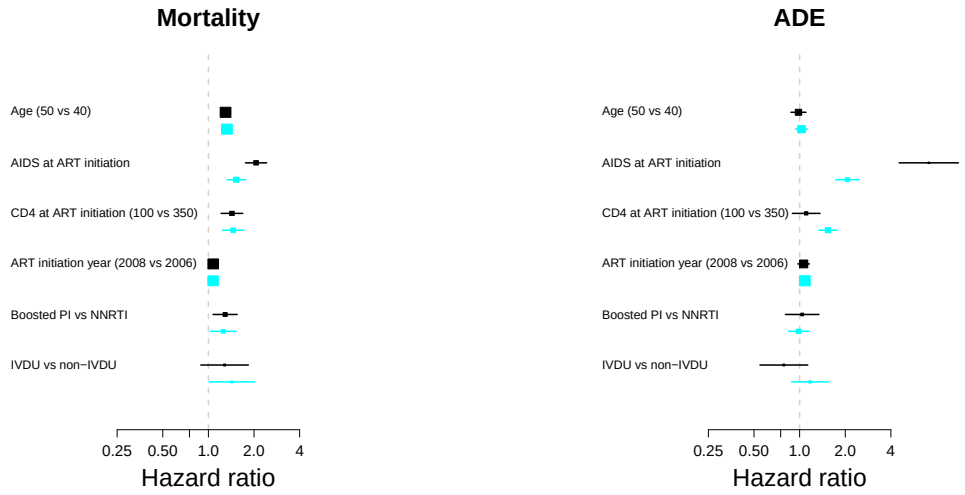


(a) Mortality

(b) ADE

Figure 2.4: Unadjusted estimates of time to mortality (a) and AIDS-defining event (b) for patients in the pre-audit (black) and post-audit (blue) datasets.

In adjusted analyses, the hazard ratios corresponding to ADE and mortality outcomes were shifted for select variables (Figure 2.5 and Figure 2.4). The hazard of death for patients with a prior history of clinical AIDS was higher in the pre-audit dataset (HR: 2.06; 95%CI: 1.76-2.42) than in the post-audit dataset (HR: 1.53; 95%CI: 1.33-1.76). The hazard of ADE for patients with a prior history of clinical AIDS was also higher in the pre-audit dataset (HR: 7.11; 95%CI: 4.54-11.15) than in the post-audit dataset (HR:2.06; 95%CI: 1.73-2.46). Hazard ratios of ADE in the post audit-dataset relative to the pre-audit dataset were higher for patients with a lower CD4 cell count (1.54; 95%CI: 1.34-1.76 vs. 1.11; 95%CI: 0.90-1.36). Differences in the hazards of death and ADE between pre-audit and post-audit datasets varied by site (results not shown).



(a) Mortality (b) ADE

Figure 2.5: Adjusted hazard ratios of mortality (a) and AIDS-defining event (b) for patients in the pre-audit (black) and post-audit (blue) datasets.

2.4 Discussion

In addition to reaffirming the findings of other studies that suggested that data audits were a useful resource to quantify the existence of errors in an observational HIV dataset (Kiragga et al. (2011); Nicol et al. (2016); Muthee et al. (2018); Puttkammer et al. (2016); Duda et al. (2012)), this study demonstrated that the results of statistical analyses and any corresponding inferences can be affected by improvements in data quality following such audits. There were many possible reasons for data entry errors, ranging from isolated errors such as typographical mistakes and misread values due to illegible handwriting to systematic issues such as misinterpreted variable definitions, miscoded value sets, or mistakes in assembling the research database. For a multi-site, multiregional consortium, the early identification and rapid resolution of such systematic issues can have a profound impact on data quality. In our data audit, we identified discrepancies between the research database and the source documents for all nine sites that participated in this study. Perhaps more interestingly, the error rates were not constant across all variables; rather, sites faced challenges with different variables. For example, high error rates in the audited dataset corresponding to dates of CD4 cell count and viral load measurements at one site uncovered a systematic error in how data entry personnel had been trained to enter this data into the study database. As a result of the audit, investigators at that site were made aware of the issue and thus were able to fix the errors in previously entered data and prevent

future incorrect entries. The audit conducted for this study was the second cycle of audits in our region. A previous audit in our region was conducted in 2008-2009 and included eight of the nine sites in this study; this was the first audit for INI-Brazil. In the initial audit cycle, data auditors noted difficulty across sites in accurately capturing ART regimen data. In general, many variables corresponding to dates were error-prone (Duda et al., 2012). While error rates corresponding to ART data were lower in both the audited and post-audit datasets from this particular cycle relative to the 2008-2009 audits, we again noted high discrepancy rates corresponding to date variables. This highlights that identifying an issue does not guarantee a resolution of an issue and that assurance of data quality must be an ongoing process. This second audit cycle was, however, the first time that the clinical endpoints data were audited at any of the sites in our consortium. Error rates for these variables tended to be higher in both the audited and post-audit datasets than error rates for variables that had been audited during the 2008-2009 audit cycle. Notably, the estimated incidence of AIDS-defining events was higher in the audited and post-audit datasets. This could be due to the nature of the variable - that clinical endpoint entries were particularly prone to errors and improper extraction by data capture personnel who lacked the necessary clinical background to identify diagnoses in paper charts. The high error rates in the clinical endpoints variable may also be indirect evidence that the audit process worked: variables that have been previously audited could be less likely to be error-prone in the next wave of audits because major errors have been identified and causes of errors have been recognized and fixed. Findings from our study suggested that discrepancy rates (relative to the pre-audit dataset) for most variables were higher in the audit database than in the post-audit database. This was not surprising: we expected more errors would be found when the auditors were actively searching for them. Variables with low audit-determined error rates would not have triggered a full source document review by the sites when preparing their post-audit databases. However, it does serve as a reminder that when conducted on a random subset of records, data audits mostly improve data quality among all patients for select variables with systematic issues and to a lesser extent the remaining variables among the audited patients. In ongoing work, we are considering statistical methods that can predict errors for unaudited patients using existing audit data and update analyses accordingly (Shepherd and Yu, 2011). Discrepancy rates for the derived variables tended to be lower relative to discrepancy rates for the primary variables in both the pre-audit versus audited dataset comparison and the pre- versus post-audit comparison. Given that the derived variables were typically composite variables of

two or more primary variables, we had anticipated that they would be more prone to error. A closer review of the data reveals that a large number of discrepancies in the primary variables were due to missingness in one of the two datasets. For variables that were collected multiple times across visits, a missing entry was often inconsequential when generating analysis variables. This reaffirmed the findings of other studies regarding the limitation of using error rates, even those focused on key variables, when making decisions regarding data quality (Mitchel et al. (2011); Smith et al. (2012)). Our study design with regard to the timing of the audit implementations did not allow us to account for temporal effects. We recognize that some corrections (e.g., entry of backlog visits) may have occurred independently from the audit process. This was a limitation of our study and we exercise caution in making conclusions regarding the long-term impact of data audits.

2.5 Conclusions

Our findings provide evidence that the SDV process can improve data quality, which can in turn have an impact on epidemiological inferences, especially for variables like the CCASAnet clinical endpoints data that had not been audited previously. We encourage the implementation of data audits for observational studies that rely on the extraction of study data from source documents, especially in multi-site settings.

2.6 Appendix A

Table 2.1: Complete list of study variables with descriptions.

Variable	Description	Source	Dataset		
			Pre-Audit	Audited	Post-Audit
birth_d	Birth date	tblBasic	Yes	Yes	Yes
male	Gender at birth (0-Female, 1-Male, 9-Unknown)	tblBasic	Yes	Yes	Yes
mode	Probable Mode of Infection	tblBasic	Yes	Yes	Yes
aids_y	AIDS dx before 1st visit	tblBasic	Yes	Yes	Yes
aids_d	AIDS dx date (if *aids_y*=1)	tblBasic	Yes	Yes	Yes
cdcstage [a]	CDC Stage at enrollment	tblBasic	Yes	Yes	No
whostage [a]	WHO Stage at enrollment	tblBasic	Yes	Yes	No
aids_cl_y [a]	Clinical AIDS dx before 1st visit	tblBasic	No	No	Yes
aids_cl_d [a]	Clinical AIDS dx date (if *aids_y*=1)	tblBasic	No	No	Yes
death_y	Did patient pass away?	tblFollow	Yes	Yes	Yes
death_d	Death date (if *death_y*=1)	tblFollow	Yes	Yes	Yes
drop_d	Date patient dropped from cohort	tblFollow	Yes	Yes	Yes
visit_d	Clinical encounter date	tblVisit	Yes	Yes	Yes
cdcstage	CDC Stage	tblVisit	Yes	Yes	Yes
whostage	WHO Stage	tblVisit	Yes	Yes	Yes
cd4_d	Date of CD4 lab test	tblLab_CD4	Yes	Yes	Yes
cd4_v	CD4 count value	tblLab_CD4	Yes	Yes	Yes
rna_d	Date of RNA lab test	tblLab_RNA	Yes	Yes	Yes
rna_v	RNA value	tblLab_RNA	Yes	Yes	Yes
art_sd	Date of ART drug start	tblART	Yes	Yes	Yes
art_ed	Date of ART drug end	tblART	Yes	Yes	Yes
art_id	Code representing ART drug(s)	tblART	Yes	Yes	Yes
ce_d	Date of clinical outcome	tblCE	Yes	Yes	Yes
base_sex	Gender at birth (0-Female, 1-Male, 9-Unknown)	Derived	Yes	Yes	Yes
base_mode	Probable Mode of Infection	Derived	Yes	Yes	Yes
base_year	Year of ART initiation	Derived	Yes	Yes	Yes
base_age	Age at ART initiation	Derived	Yes	Yes	Yes
base_ADE	Clinical AIDS at ART initiation	Derived	Yes	Yes	Yes
base_ARTregimen	1st ART regimen	Derived	Yes	Yes	Yes
base_CD4	CD4 cell count at ART initiation	Derived	Yes	Yes	Yes
base_nadirCD4	Lowest CD4 cell count prior to ART initiation	Derived	Yes	Yes	Yes
base_VL	Viral load at ART initiation	Derived	Yes	Yes	Yes
base_VLundetectable	Undetectable VL at ART initiation	Derived	Yes	Yes	Yes
event_death	Did patient pass away?	Derived	Yes	Yes	Yes
event_ADE	Did patient have post-ART clinical outcome?	Derived	Yes	Yes	Yes
time_follow	Time from ART initiation to death/censoring	Derived	Yes	Yes	Yes
time_ADE	Time from ART initiation to ADE/censoring	Derived	Yes	Yes	Yes

The data protocol further defining variables is available at https://www.ccasanet.org/resources/.
[a] In 2014, the data transfer protocol pertaining to clinical AIDS status at enrollment was updated. Certain variables were deprecated in tblBasic and replaced with new variables. These variables are not included in comparisons of pre-audit and post-audit datasets.

Table 2.2: Overview of audit frequency by site.

Site	Audited N(%)	Not Audited N(%)	Total
HF-Argentina	31 (89%)	4 (11%)	35
CMH-Argentina	12 (57%)	9 (43%)	21
INI-Brazil	31 (52%)	29 (48%)	60
FA-Chile	34 (97%)	1 (3%)	35
GHESKIO-Haiti	28 (80%)	7 (20%)	35
IHSS-Honduras	30 (100%)	0 (0%)	30
HE-Honduras	30 (100%)	0 (0%)	30
INNSZ-Mexico	24 (69%)	11(31%)	35
IMTAvH-Peru	30 (86%)	5 (14%)	35
Overall	250	66	316

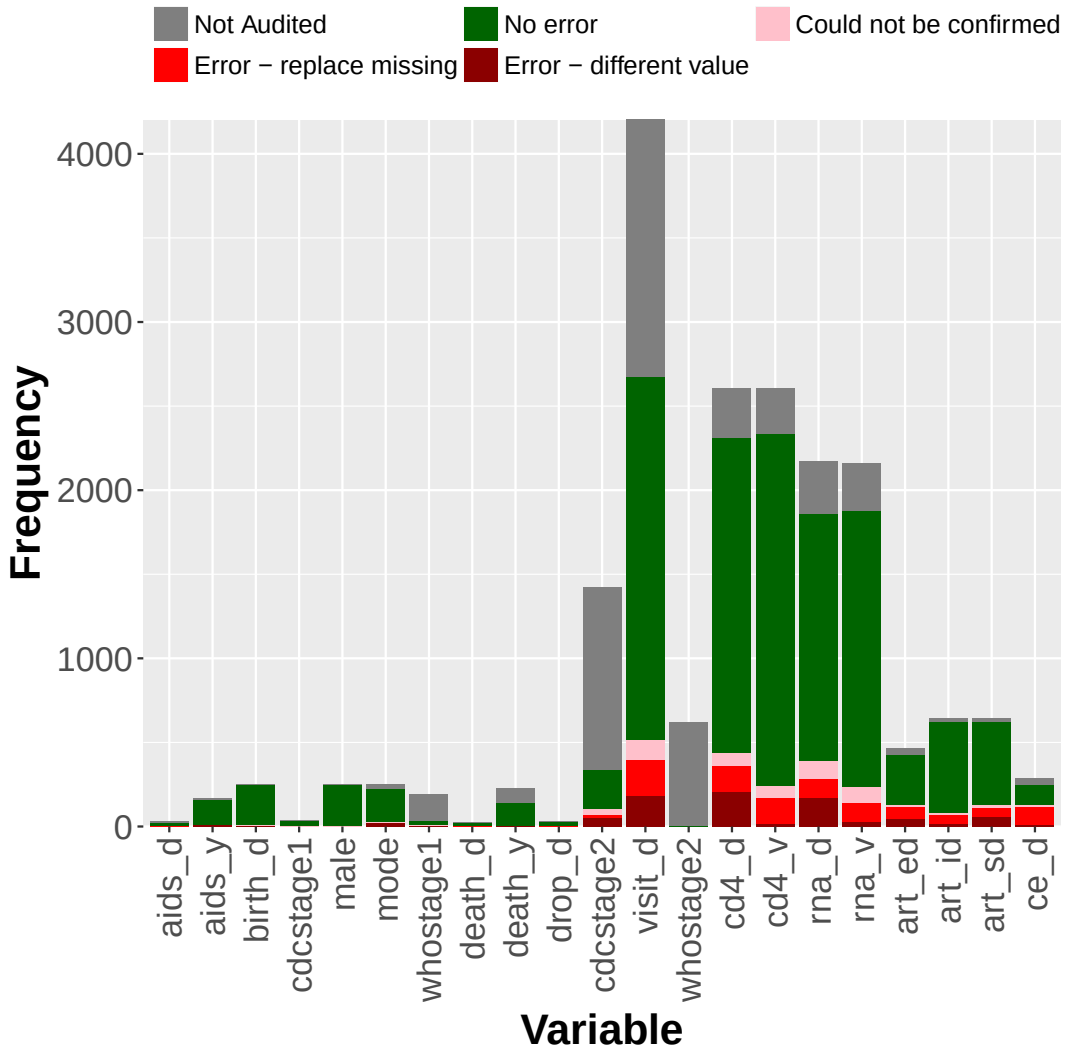
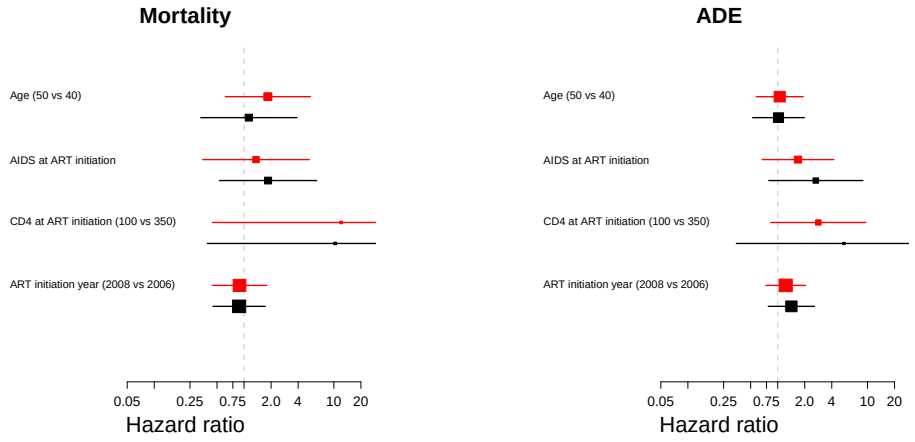


Figure 2.6: Summary of reported audit findings for all audited variables.

Table 2.3: Auditing results for each variable entry.

Form	Variable	Total Entries	Not audited	Total Audited	No error A1	Error - diff value A2 + A3	Error - replace NA A4	Cannot be confirmed A5	Error Rate (A2+A3+A4+A5)/ (A1+A2+A3+A4+A5)
basic									
	male	250	5	245	242	2	0	1	1.2%
	cdcstage1	35	0	35	34	1	0	0	2.9%
	birth.d	250	4	246	237	7	0	2	3.7%
	aids.y	164	6	158	146	9	1	2	7.6%
	mode	250	28	222	192	23	0	7	13.5%
	aids.d	28	6	22	18	1	3	0	18.2%
	whostage1	192	157	35	27	7	0	1	22.9%
follow									
	death.y	226	88	138	131	5	0	2	5.1%
	drop.d	30	0	30	25	0	4	1	16.7%
	death.d	26	2	24	18	0	4	2	25.0%
visit									
	whostage2	623	616	7	7	0	0	0	0.0%
	visit.d	4203	1529	2674	2157	183	211	123	19.3%
	cdcstage2	1425	1084	341	232	54	16	39	32.0%
cd4									
	cd4.v	2607	273	2334	2090	17	155	72	10.5%
	cd4.d	2607	292	2315	1875	205	155	80	19.0%
rna									
	rna.v	2163	285	1878	1644	27	116	91	12.5%
	rna.d	2175	314	1861	1471	173	113	104	21.0%
art									
	art.id	644	18	626	546	17	56	7	12.8%
	art.sd	645	21	624	496	56	57	15	20.5%
	art.ed	467	36	431	298	46	73	14	30.9%
ce									
	ce.d	286	36	250	124	10	108	8	50.4%
	overall	19296	4800	14496	12010	843	1072	571	17.1%

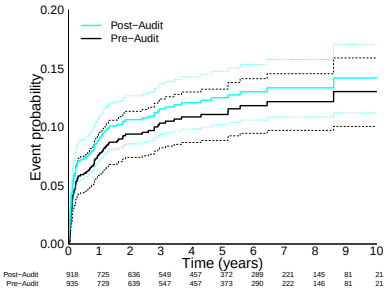


(a) Mortality (b) ADE

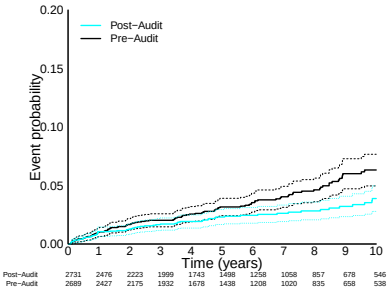
Figure 2.7: Adjusted hazard ratios of mortality (a) and AIDS-defining event (b) for patients in the pre-audit (black) and audited (red) datasets.

Table 2.4: Adjusted hazard ratios of mortality and AIDS-defining event for all patients enrolled at time of data audit using the pre-audit and post-audit datasets.

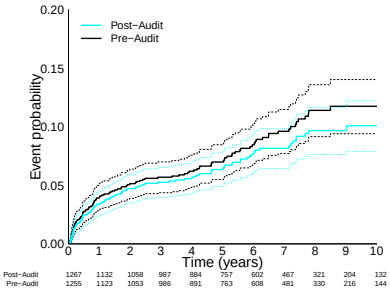
	Mortality		ADE	
	Pre-audit Hazard Ratio	Post-audit Hazard Ratio	Pre-audit Hazard Ratio	Post-audit Hazard Ratio
Gender				
Female	Ref	Ref	Ref	Ref
Male	1.07 (0.95 - 1.20)	1.05 (0.93 - 1.18)	0.90 (0.79 - 1.03)	0.95 (0.84 - 1.07)
Age				
20	1.09 (0.85 - 1.40)	1.00 (0.77 - 1.29)	1.19 (0.94 - 1.51)	1.05 (0.86 - 1.28)
30	0.92 (0.82 - 1.04)	0.89 (0.79 - 1.00)	1.05 (0.92 - 1.19)	0.99 (0.88 - 1.11)
40	Ref	Ref	Ref	Ref
50	1.30 (1.21 - 1.39)	1.33 (1.24 - 1.42)	0.98 (0.88 - 1.10)	1.03 (0.94 - 1.13)
60	1.77 (1.49 - 2.11)	1.83 (1.54 - 2.18)	0.97 (0.73 - 1.28)	1.06 (0.83 - 1.34)
Clinical AIDS				
No	Ref	Ref	Ref	Ref
Yes	2.06 (1.76 - 2.42)	1.53 (1.33 - 1.76)	7.11 (4.54 - 11.15)	2.06 (1.73 - 2.46)
Nadir CD4				
50	1.86 (1.56 - 2.22)	1.93 (1.62 - 2.30)	1.22 (0.96 - 1.54)	1.86 (1.60 - 2.16)
100	1.43 (1.21 - 1.69)	1.46 (1.24 - 1.71)	1.11 (0.90 - 1.36)	1.54 (1.34 - 1.76)
200	1.04 (0.91 - 1.18)	1.04 (0.92 - 1.18)	0.96 (0.84 - 1.08)	1.14 (1.05 - 1.25)
350	Ref	Ref	Ref	Ref
Initiation Year				
2000	0.83 (0.70 - 0.99)	0.83 (0.69 - 0.98)	0.98 (0.82 - 1.18)	1.14 (0.97 - 1.34)
2002	0.84 (0.73 - 0.96)	0.82 (0.71 - 0.94)	0.92 (0.78 - 1.07)	1.06 (0.92 - 1.23)
2004	0.89 (0.82 - 0.98)	0.87 (0.79 - 0.95)	0.94 (0.86 - 1.02)	1.01 (0.93 - 1.08)
2006	Ref	Ref	Ref	Ref
2008	1.08 (1.00 - 1.16)	1.07 (1.00 - 1.15)	1.06 (0.97 - 1.16)	1.08 (1.01 - 1.16)
2010	1.03 (0.84 - 1.26)	0.95 (0.78 - 1.17)	1.12 (0.90 - 1.39)	1.24 (1.05 - 1.47)
2012	0.93 (0.61 - 1.41)	0.78 (0.50 - 1.19)	1.17 (0.81 - 1.69)	1.46 (1.09 - 1.95)
ARV Class				
NNRTI	Ref	Ref	Ref	Ref
Boosted PI	1.29 (1.07 - 1.55)	1.25 (1.03 - 1.52)	1.04 (0.81 - 1.34)	0.98 (0.85 - 1.15)
Other	1.09 (0.88 - 1.34)	1.15 (0.93 - 1.42)	1.02 (0.82 - 1.28)	1.10 (0.92 - 1.31)
IVDU				
No	Ref	Ref	Ref	Ref
Yes	1.28 (0.89 - 1.83)	1.43 (1.01 - 2.02)	0.79 (0.55 - 1.13)	1.17 (0.88 - 1.56)



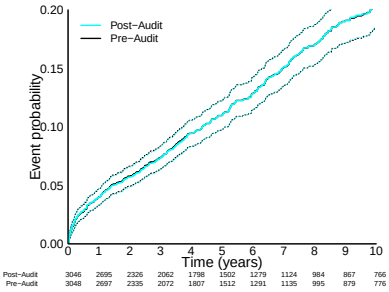
(a) Site 1



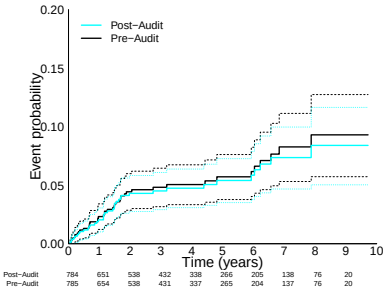
(b) Site 2



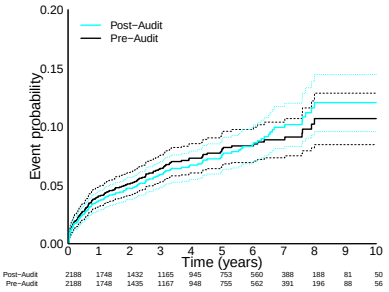
(c) Site 3



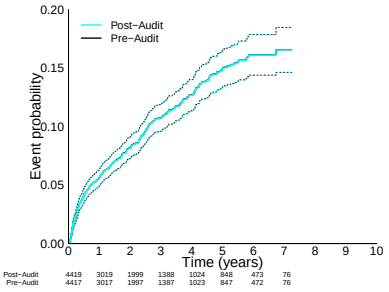
(d) Site 4



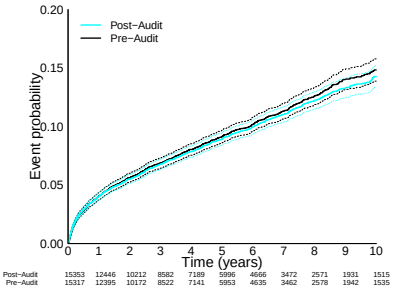
(e) Site 5



(f) Site 6

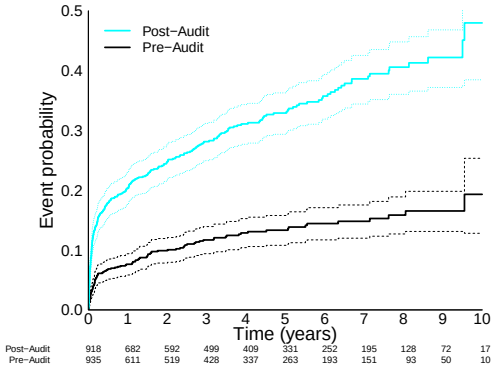


(g) Site 7

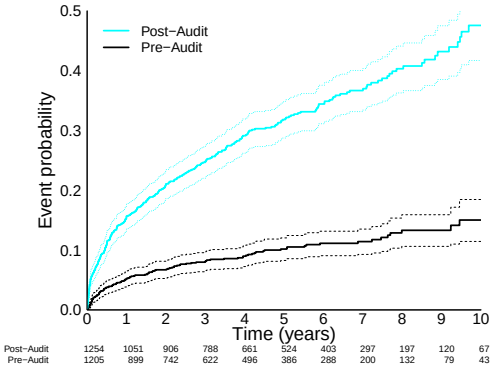


(h) Overall

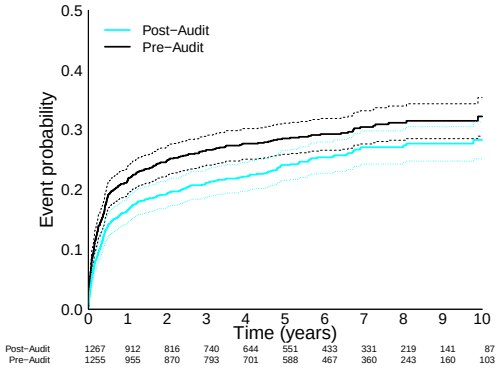
Figure 2.8: Estimated cumulative incidence of death by site for patients in the pre-audit (black) and post-audit (blue) datasets.



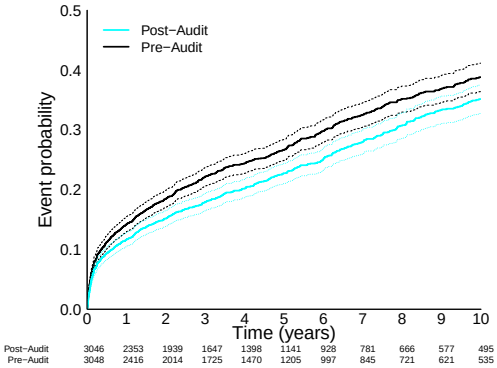
(a) Site 1



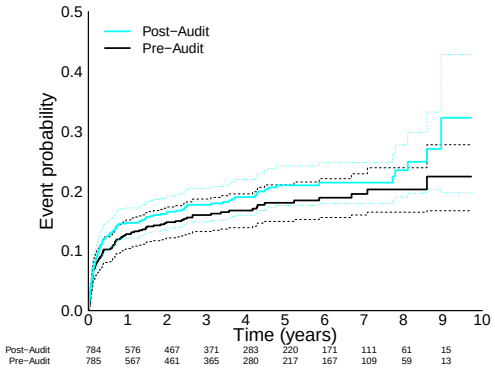
(b) Site 2



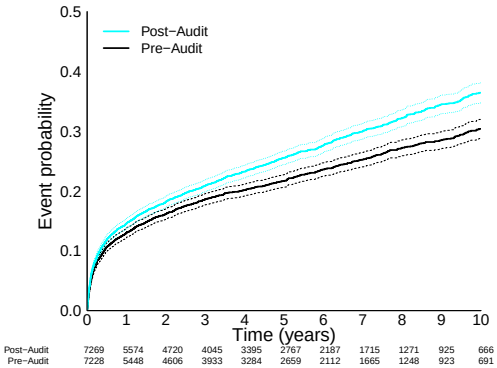
(c) Site 3



(d) Site 4



(e) Site 5



(f) Overall

Figure 2.9: Estimated cumulative incidence of ADE by site for patients in the pre-audit (black) and post-audit (blue) datasets.

CHAPTER 3

ACCOUNTING FOR DEPENDENT ERRORS IN PREDICTORS AND TIME-TO-EVENT OUTCOMES USING ELECTRONIC HEALTH RECORDS, VALIDATION SAMPLES, AND MULTIPLE IMPUTATION

3.1 Introduction

Routinely collected electronic health record (EHR) data are increasingly being used for medical research. An alarming number of studies, however, are raising concerns regarding the quality of EHR data and consequently misleading findings (e.g., Chan et al. (2010); Floyd et al. (2012); Duda et al. (2012)). Some errors, such as values falling outside specific ranges (e.g., negative CD4 counts or date of death before enrollment), can be identified using computerized data quality checks and flagged for chart review. Other errors are harder to detect with simple queries. For example, the date of treatment initiation may be incorrectly recorded, but the documented date is within the follow-up period. Furthermore, the existence and magnitude of errors may be correlated across multiple variables. For example, if the treatment initiation date is incorrect then lab values at the time of treatment initiation and the calculated time from initiation to some event are also likely incorrect. To identify such errors, all relevant error-prone variables would have to be verified; however, such a resource-intensive process may not be feasible in settings with limited funding.

An alternative to verification of all records is to perform data audits or validation in a subset of records. This is generally done by selecting a random set of records and verifying data accuracy for key variables. If non-trivial error rates are revealed, one might remove the data in question or re-enter all data. These options, however, seem unsatisfactory. A more appealing option would be to incorporate the audit or validation data into the analysis.

The data available following an audit – an error-prone measurement for all records and a "gold standard" measurement for a subset of records – resembles the data one might need to correct for measurement error. While the statistical literature regarding measurement error is substantial, most methods involving time-to-event data focus only on covariate measurement error. These methods include regression calibration (Prentice, 1982), corrected score methods (Nakamura, 1990; Huang and Wang, 2000), conditional score methods (Tsiatis and Davidian, 2001), and SIMEX (Cook and Stefanski, 1994; Li and Lin, 2003). There have also been select studies related to time-to-event outcome measurement error, with methods corresponding to

errors in event indicators (Richardson and Hughes, 2000; Magaret, 2008; Hunsberger et al., 2010) or the failure time (Skinner and Humphreys, 1999; Korn et al., 2010; Oh et al., 2018). While correlated errors in covariates and uncensored outcomes have been previously considered (Shepherd and Yu, 2011; Shepherd et al., 2012), no existing methods address situations with errors in both the covariates and the time-to-event outcome. Given that errors in EHR data typically occur across multiple variables and these errors are generally correlated, the current measurement error literature is not equipped to handle such multidimensional errors seen in practice.

Measurement error with a validation subsample can also be thought of as a missing data problem (Little and Rubin, 2014). Specifically, the complete data are known for all records that were audited, and the true values of certain variables are missing in records that were not audited. Because researchers typically randomly select which records to audit, the missing data mechanism is generally missing at random and standard methods for addressing missing data could be applicable. For example, a multiple imputation (MI) approach could be employed by fitting models using the complete data and imputing missing values for unvalidated records. This approach has been implemented in previous studies for various measurement error scenarios, including mismeasured binary (Edwards et al., 2013) and continuous (Shepherd et al., 2012) outcomes as well as measurement error in the exposure of a time-to-event outcome (Cole et al., 2006). Although relevant, none of this work considers the situation where there are errors, likely correlated, in both predictors and the time-to-event outcome. Furthermore, it does not address errors in indicators of patient eligibility that determine whether a patient should be included in the analysis. These situations, common in practical applications of EHR data for analyses, add considerable complexity when attempting to address measurement error.

Our goal is to obtain unbiased and efficient estimates in time-to-event analyses using error-prone EHR data, a subsample of which has been validated. By unbiased estimates, we mean estimates that are unbiased for what we would have obtained had we validated the entire dataset. (Of course, bias due to other sources, e.g., informative censoring or unmeasured confounding, may still remain after data validation. Ideally, our methods will be flexible enough to be used with other methods for incorporating strategies to deal with these other sources of bias.) Such methods could have a large impact on the practical use of EHR data as they provide a means to obtaining valid estimates while reducing research costs. In this manuscript, we describe and implement a multiple imputation-based strategy to account for correlated errors in both predictors and time-to-event outcomes as well as analysis eligibility. We illustrate

our approach using EHR data from a large HIV clinic, the Vanderbilt Comprehensive Care Clinic (VCCC). The VCCC has validated all key research variables for all of their patients, and they have preserved datasets both before and after this validation. The fully validated data revealed errors in the original data (illustrated in Section 3.2), and can be used to see how biased estimates would have been had the data not been validated. In addition, since we have both unvalidated and fully validated data for all patient records, this dataset is ideal for examining the performance of our proposed approach.

In Section 3.2, we present our motivating example: we describe the VCCC cohort, illustrate errors in the data, and demonstrate the bias in estimates using unvalidated data. In Section 3.3, we formalize our problem and present our strategy for obtaining improved estimates after partial data validation. In Section 3.4, we present results of our approach, first using a single, randomly sampled validation subset of $n = 1000$ VCCC records, and second across multiple randomly sampled subsets. We are able to estimate the mean squared error of our approach as a function of validation sample size, and to compare it to analyses that are limited to only the validated subsample. In Section 3.5, we use a simulation study to evaluate the peformance of the TDMI approach when the imputation model is misspecified compared to a fully specified imputation model. In Section 3.6, we discuss our results and suggest areas for future research. All analyses described below were performed using R Version 3.2. Analysis code is posted at http://biostat.mc.vanderbilt.edu/ArchivedAnalyses.

3.2 Motivating example

In this study, we analyzed data on 4217 HIV-positive patients who established care at the VCCC between 1998 and 2011. Briefly, the VCCC is an outpatient clinic that provides primary and subspecialty care for persons living with HIV. As part of routine treatment and care, data relevant to the patient's clinical experience were collected over time. This included demographic characteristics as well as laboratory measurements, pharmacy dispensations, opportunistic infections, and vital status. Data before enrollment was also recorded, usually during the initial visit, based on patient recall and outside medical records. The median length of follow-up after enrollment was 3.2 years (IQR: 1.1-6.8). The majority of patients were male (76%) and the median age at enrollment was 38 years (IQR: 31-45).

Data at the VCCC was collected and electronically recorded by health care providers, typically nurses and physicians. Research protocols mandated that chart reviews were

performed for all VCCC records to validate key variables. A team of data abstractors performed the data validation. After this comprehensive chart review process, two datasets were available. The first dataset, which we refer to as the unvalidated dataset, contained the values entered for all 4217 records prior to the chart review. The second dataset, which we refer to as the validated dataset, contained the recorded values for the same 4217 records after thorough chart review. Throughout this study we consider the validated dataset to be correct.

For this study, we considered the association between CD4 count at time of antiretroviral therapy (ART) initiation and the time from ART initiation until first AIDS-defining event (ADE). Specifically, we calculated the incidence of ADE using Kaplan-Meier methods and the hazard ratio (HR) for a 100 cell/mm^3 increase in CD4 count using a univariate Cox proportional hazards regression model. All patients included in the analysis cohort were adults ($\geq$ 18 years) who had initiated ART as adults. Patients were excluded if they started ART prior to enrollment, had an indeterminate ART start date, or had a documented ADE prior to ART initiation. These inclusion and exclusion criteria are common for HIV studies.

We performed the same statistical analysis for both the unvalidated and validated datasets. The incidence of ADE was higher in the unvalidated dataset across the entire study period. The estimated incidence of ADE at 5 years was 13.6% (95% confidence interval [CI]: 11.4% - 15.7%) for the unvalidated dataset and 8.3% (95% CI: 6.6% - 10.0%) for the validated dataset. A 100 cell/mm^3 increase in CD4 count was associated with a much weaker decrease in the hazard of ADE in the unvalidated dataset (HR: 0.80; 95%CI: 0.74 - 0.86) compared to the validated dataset (HR: 0.63; 95%CI: 0.55 - 0.72).

There were many discrepancies between the unvalidated and validated datasets. In the unvalidated dataset, 1764 patients satisfied the criteria for inclusion in the analysis cohort. In the validated dataset, 1601 patients met all inclusion criteria. A total of 1409 met the inclusion criteria for both analysis cohorts, suggesting 547 (13%; 355 wrongly included and 192 wrongly excluded) patients were incorrectly classified in the unvalidated dataset. Among those patients that met inclusion criteria for both analysis cohorts, the variables indicating ADE status were discordant for 132 (9%) patients; 118 patients were re-classified as not having an ADE and treated as censored at the time of their last follow-up visit while 14 patients were re-classified as having an ADE in the validated analysis cohort. The time from ART initiation to an ADE or end of study was incorrectly recorded for 447 (32%) patients, with a median of 1 days late and an interquartile range (IQR) of 366 days early to 21 days late. The baseline

Table 3.1: Comparison of variables in the unvalidated and validated datasets among the 4217 patients.

	Notation (see Section 3)	Discrepancy magnitude n or median(IQR)
All patients		4217
Different ART start date	S_0	1745 (41.4%)
Discrepancy in ART start dates (days)	U_0	14 (-222, 37)
Different ADE date	S_E	1223 (29.0%)
Discrepancy in ADE date (days)	U_E	-14 (-5, 165)
Met inclusion criteria in both datasets	$W = 1, W^* = 1$	1409
Different ADE status	$D \neq D^*$	132 (9.4%)
ADE in unvalidated, no ADE in validated	D^*=1, D=0	118
ADE in validated, no ADE in unvalidated	D^*=0, D=1	14
Different time from ART initiation to ADE	$Y \neq Y^*$	447 (31.7%)
Discrepancy in time from ART to ADE (days)	$Y - Y^*$	1 (-366, 21)
Different baseline CD4 count	$X_1 \neq X_1^*$	76
Different baseline CD4 (diff ART start date)	$S_0 = 1, S_{X_1} = 1$	76
Discrepancy in baseline CD4 count	$U_{X_{1,0}}$	22 (-23, 88)
Different baseline CD4 (same ART start date)	$S_0 = 0, S_{X_1} = 1$	0
Discrepancy in baseline CD4 count	U_{X_1}	–

Abbreviations: ADE, AIDS-defining event. ART, antiretroviral therapy. IQR, interquartile range.

CD4 count was also incorrect for 76 (5%) patients, with a median discrepancy of 22 cells/mm^3 too high and an IQR of 23 cells/mm^3 too low and 88 cells/mm^3 too high. Table 3.1 includes further details comparing the unvalidated and validated datasets.

As with most studies using EHR data, the variables used in our analyses were primarily derived variables (e.g., baseline CD4 was determined by identifying the laboratory measurement in one table with a date closest to the first ART initiation date in a separate table). Discrepancies in derived variables were mostly due to errors in the indicators and dates of ART and ADE. Among all 4217 patients, there were 1745 (41%) patients with an incorrect ART start date and 1253 (30%) patients with an incorrect ADE (or end of follow-up) date. All discrepancies in the baseline CD4 count were due to discrepancies in the ART start date.

3.3 Our approach

In the previous section, we showed that estimates using just the unvalidated dataset were markedly biased. These findings highlight, at least in our setting, the importance of validating EHR data. Our goal in this study is to obtain low bias and low variance estimates after validating only a subsample of the EHR. In this section,

we formalize the problem analytically and describe our analysis approach.

3.3.1 Notation

Let T_B denote the date of enrollment, T_0 the date of ART initiation and T_E the date of first ADE. Since some patients may not have an ADE before the end of follow-up, let T_C denote the last follow-up ("end of study") date. Using these dates, we derive the variables corresponding to the outcome: the time from ART initiation to ADE or end of study, $Y = min(T_C, T_E) - T_0$, and an indicator of an ADE, $D = I(T_E \leq T_C)$.

Let $X(t) = (X_1(t), X_2(t), ..., X_p(t))$ denote a vector of p covariates for a patient on a given date, t. For example, let $X_1(t)$ denote CD4 count on date t. While this notation allows each covariate to change values over time, we note that some covariates may be time-invariant. Since we are interested in values at time of ART initiation (T_0), we define a vector of "baseline" variables as $X = X(T_0) = (X_1(T_0), X_2(T_0), ..., X_p(T_0))$, where $X_1(T_0)$ corresponds to baseline CD4 count and $X_2(T_0), ..., X_p(T_0)$ correspond to the remaining baseline covariate values. Finally, let W denote whether a patient was included in the analysis cohort. Patients were included if they started ART after enrollment ($T_B \leq T_0 < T_C$) and if they did not have an ADE before starting ART ($T_0 < T_E$). The quadruplet (W, X, Y, D) represents the data for our time-to-event analyses from the validated records (i.e. the gold standard).

In our specific application, among those meeting inclusion criteria ($W = 1$) we are interested in estimating the probability of an event at time t, $P(T_E - T_0 \leq t)$ and the hazard ratio in the proportional hazards model, $\lambda(t|X) = \lambda_0(t)exp(\beta X)$. In the absence of measurement error (and assuming no other sources of bias, such as informative censoring), these can be consistently estimated using Kaplan-Meier and Cox regression methods.

Since patient records are potentially error-prone, we use separate notation for the data from the unvalidated records. Let the unvalidated time of ART initiation be $T_0^* = T_0 + S_0U_0$ where S_0 is the indicator of an error and U_0 is the magnitude of the error. Similarly, let the unvalidated time of ADE be defined as $T_E^* = T_E + S_EU_E$ and define the end of study date in the unvalidated dataset as $T_C^* = T_C + S_CU_C$. Let the date-specific vector of p covariates in the unvalidated dataset be $X^*(t) = X(t) + S_{X(t)}U_{X(t)}$, where $S_{X(t)}$ is a vector of length p with indicators of errors for covariates on date t and $U_{X(t)}$ is the corresponding magnitude of those errors.

The derived variables corresponding to the outcome in the unvalidated dataset are

$D^* = I(T_E^* \leq T_C^*)$ and $Y^* = min(T_E^*, T_C^*) - T_0^*$. The unvalidated baseline predictor variables are defined as $X^* = X^*(T_0^*) = X(T_0 + S_0 U_0) + S_{X(T_0^*)} U_{X(T_0^*)}$. Let W^* be an indicator for inclusion, defined as $I(T_B \leq T_0^* < T_C^*) I(T_0^* < T_E^*)$. We denote the unvalidated data used for analyses as the quadruplet (W^*, X^*, Y^*, D^*).

Note that our model to this point makes no assumptions regarding the distribution or the correlation of the errors. We acknowledge that the error indicators and magnitudes may be highly dependent, as based on our experience, those records with errors in one variable are more likely to have errors in another variable, particularly when there are derived variables. For simplicity, we have presented error terms as additive, but they may also be written more generally (e.g., $T_0^* = g(T_0, U_0, S_0)$; for instance, $g(T_0, U_0, S_0) = T_0 U_0^{S_0}$ could be used to imply a multiplicative model).

Finally, let $V = 1$ denote that data validation was performed for all variables. For those records with $V = 1$, we have (W^*, X^*, Y^*, D^*) and (W, X, Y, D), whereas for those records with V=0 we have only (W^*, X^*, Y^*, D^*). In our VCCC example, $V = 1$ for all records, so an analyst would ignore the error-prone unvalidated data and draw inference using only the validated data. We will consider the situation where $V = 1$ for only a subsample of patients.

Note that under certain conditions regarding errors between unvalidated and validated datasets, we could draw on existing statistical methods for correcting measurement error. As highlighted in the Introduction, there has been substantial work for time-to-event outcome studies regarding covariate measurement error (X^*, Y, D) and, to a lesser degree, measurement error in the event indicator (X, Y, D^*) or time-to-event (X, Y^*, D). Methods to address correlated errors in covariates and uncensored outcomes (X^*, Y^*) have also been considered. However, methods for simultaneously dealing with errors in predictors, event indicators, and times-to-event (X^*, Y^*, D^*) have not been considered, and because of potential dependence between these errors, it is not possible to simply sequentially apply existing methods. Furthermore, for our motivating example, we also need to consider errors with the inclusion criteria (W^*, X^*, Y^*, D^*).

3.3.2 Multiple Imputation: model fitting and time-discretization

Our strategy is to approach this as a missing data problem where the quadruplet (W^*, X^*, Y^*, D^*) is available for all records and the quadruplet (W, X, Y, D) is missing for those with $V = 0$. This requires the construction of a model for the joint distribution of (W, X, Y, D) conditional on (W^*, X^*, Y^*, D^*). The model will be fit

using a subsample of records with $V = 1$; values for the remaining records ($V = 0$) will be imputed using these fitted models. Therefore, the primary challenge is obtaining adequate models.

Consider the factorization of the distribution of (W, X, Y, D) conditional on the quadruplet (W^*, X^*, Y^*, D^*):

$$f(W, X, Y, D|W^*, X^*, Y^*, D^*) = f(W|W^*, X^*, Y^*, D^*)\ f(X|W, W^*, X^*, Y^*, D^*) \quad \times$$
$$f(Y|W, X, W^*, X^*, Y^*, D^*)\, f(D|W, X, Y, W^*, X^*, Y^*, D^*), \tag{3.1}$$

where $f(\cdot)$ denotes a generic probability density/mass function. With this factorization, each component of (3.1) could be fit separately using existing data from the subsample of records with $V = 1$. However, these models would be constructed using derived variables that are functions of other error-prone variables, $(T_0, T_E, T_C, X(t))$, likely making their predictive ability poor. Furthermore, it may be difficult to incorporate informative time-varying covariates into these models. For example, a marked drop in viral load is strong evidence that someone has begun treatment, but it is unclear how to incorporate such information into models with these derived variables.

To more closely approximate the error structure of the data, a more appropriate strategy might be to directly model the original variables $(T_0, T_E, T_C, X(t))$ given error prone values $(T_0^*, T_E^*, T_C^*, X^*(t))$. The factorization of the distribution of $(T_0, T_E, T_C, X(t))$ conditional on $(T_0^*, T_E^*, T_C^*, X^*(t))$ is still challenging. Modeling the ART start date and first ADE date may require strong distributional assumptions that account for many patients not experiencing an event. Furthermore, one of the variables to be modeled, $X(t)$, contains time-varying covariates, where the number of observations and the timing of observations varies per patient. This requires a model that accounts for the distribution of these observations over time as well as conditions on them in the models for the other variables.

An alternative strategy, which we adopt here and refer to as time-discretized modeling, divides time into intervals (e.g., days, months, years) and assesses values for variables during each interval. This approach employs a well-known strategy for modeling time-to-event data using pooled logistic regression (D'Agostino et al., 1990). Similar approaches have been implemented with marginal structural models (Hernán et al., 2001) and ecological statistics (Turchin, 1998; McClintock et al., 2014), where discretization is used to allow for time-varying covariates and to reduce

computationally-intensive tasks.

Here, variables are divided into monthly intervals, indexed by m, since the date of enrollment ($m = 0$). Specifically, let $\mathcal{A}_m$ be an indicator for a patient initiating at least one different ART drug during month m; if a patient is not on ART or continues the same ART regimen as the previous month, they are assigned $\mathcal{A}_m = 0$. Let $\mathcal{D}_m$ be an indicator of an ADE occurring during month m. Let $\mathcal{X}_m$ correspond to the most recent covariate values observed during month m, and finally, let $\mathcal{C}_m$ be an indicator that the last follow-up visit for a patient occurred during month m.

Let $\overline{\mathcal{A}} = \{\mathcal{A}_0, \mathcal{A}_1, ..., \mathcal{A}_{M_{post}}\}$ designate the complete set of monthly new ART drug indicators in the validated dataset, where M_{post} denotes the longest possible length of follow-up (in months) among all patients. The variables $\overline{\mathcal{D}}$, $\overline{\mathcal{C}}$, and $\overline{\mathcal{X}}$ are similarly defined. For the unvalidated dataset, we have $\overline{\mathcal{A}^*}$, $\overline{\mathcal{D}^*}$, $\overline{\mathcal{C}^*}$, and $\overline{\mathcal{X}^*}$. This notation implicitly assumes that the date of enrollment is correct in the unvalidated dataset; this assumption is met in the VCCC dataset, but could be relaxed by using some other date to anchor time.

With this framework, we can construct a model for the joint distribution of the variables in the validated dataset $(\overline{\mathcal{A}}, \overline{\mathcal{D}}, \overline{\mathcal{C}}, \overline{\mathcal{X}})$ conditional on the distribution of the variables in the unvalidated dataset $(\overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}, \overline{\mathcal{X}^*})$ by decomposing it into separate components:

$$\begin{aligned} f(\overline{\mathcal{X}}, \overline{\mathcal{A}}, \overline{\mathcal{D}}, \overline{\mathcal{C}} | \overline{\mathcal{X}^*}, \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}) = & f(\overline{\mathcal{X}} | \overline{\mathcal{X}^*}, \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}) \times \\ & f(\overline{\mathcal{A}} | \overline{\mathcal{X}}, \overline{\mathcal{X}^*}, \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}) \times \\ & f(\overline{\mathcal{D}} | \overline{\mathcal{A}}, \overline{\mathcal{X}}, \overline{\mathcal{X}^*}, \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}) \times \\ & f(\overline{\mathcal{C}} | \overline{\mathcal{D}}, \overline{\mathcal{A}}, \overline{\mathcal{X}}, \overline{\mathcal{X}^*}, \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}) \end{aligned} \tag{3.2}$$

With this decomposition, we directly model discretized versions of the original variables that are in error, rather than downstream, derived variables. By incorporating error-prone and corrected variables in models, we account for potential dependencies in errors across variables. Time-varying covariates are also easier to incorporate. For example, the probability of starting a new ART regimen in a given month, $\mathcal{A}_m$, can be modeled conditional on the unvalidated indicator of starting a new ART regimen for that month, $\mathcal{A}^*_m$, and time-varying covariates $\overline{\mathcal{X}}$ such as viral load prior to, during, and after month m. Specific implementation details are given in the next section.

Component models can be fit using the records with validated data (i.e., those with

$V = 1$) using appropriate methods (e.g., binary variables can be modeled using logistic regression). Each person-month is treated as an independent observation. When predicting values for the remaining records (i.e., those with $V = 0$), we account for the uncertainty in the prediction model using a multiple imputation procedure. First, we draw an independent sample of the parameter estimates of the fitted models (e.g., we sample from a multivariate normal distribution with the mean as the parameter estimates and variance as the variance-covariance matrix of the parameter estimates). Using $(\overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}, \overline{\mathcal{X}^*})$ and these randomly drawn parameter estimates, we impute values of $(\overline{\mathcal{A}}, \overline{\mathcal{D}}, \overline{\mathcal{C}}, \overline{\mathcal{X}})$ for all records with $V = 0$. From these imputed values of $(\overline{\mathcal{A}}, \overline{\mathcal{D}}, \overline{\mathcal{C}}, \overline{\mathcal{X}})$, we then derive imputed values for the variables used in our analyses, denoted as $(\widehat{W}, \widehat{X}, \widehat{Y}, \widehat{D})$. With this step, we "undiscretize" the imputed values; values are converted to the unit of measurement of the original variables using a fixed conversion. For example, an imputed ART initiation 2 months after enrollment would be reported as 60 days after enrollment. Next, we generate a complete dataset consisting of the true, observed values of the audited records and the predicted values of the unaudited records, denoted as

$$(W^{comp}, X^{comp}, Y^{comp}, D^{comp}) = \begin{cases} (\widehat{W}, \widehat{X}, \widehat{Y}, \widehat{D}) & \text{if } V = 0 \\ (W, X, Y, D) & \text{if } V = 1 \end{cases}$$

We then repeat the process of randomly sampling parameter estimates, predicting values, and combining datasets, until we have B complete datasets. Here, B is the number of imputations performed. For each of the B complete datasets, we obtain estimates using Kaplan-Meier and Cox regression methods. The parameter estimates from these procedures are then averages across iterations. To properly account for uncertainty in the setting of incompatible imputation and analysis models, we use the multiple imputation variance estimator proposed by Robins and Wang (2000) to calculate confidence intervals.

For this time-discretized modeling and imputation (TDMI) approach to yield unbiased estimates in large samples, standard assumptions for the validity of multiple imputation must be met (Schafer, 1999). The missing at random assumption can be translated to our application as $V \perp\!\!\!\perp (\overline{\mathcal{A}}, \overline{\mathcal{D}}, \overline{\mathcal{C}}, \overline{\mathcal{X}}) | (\overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}, \overline{\mathcal{X}^*})$, or that conditional on the observed data, selection for validation (V) is independent of the correct values. Another key assumption is that the imputation model, $f(\overline{\mathcal{X}}, \overline{\mathcal{A}}, \overline{\mathcal{D}}, \overline{\mathcal{C}} | \overline{\mathcal{X}^*}, \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*})$, is properly specified. The TDMI approach handles differential measurement error through the imputation model (i.e., it requires no assumption of non-differential measurement error), but covariates associated with differential error need to be correctly

included in the model. Finally, because we are estimating parameters defined on a continuous time scale after imputing data from models on a discrete time scale, we assume that the discrete time scale is a good approximation to the continuous time scale, which has been seen by others to be the case as long as the unit of discretized time is not too coarse (e.g., D'Agostino et al. (1990)).

3.3.3 Implementation details

In this section, we highlight key simplifications and noteworthy specifications that were made in our application of the TDMI to EHR data from the VCCC. Full model details are in Appendix B.

First, the end of study date for each patient did not vary between the unvalidated and validated records and thus we did not need to model $\overline{\mathcal{C}}$ as it was perfectly predicted by $\overline{\mathcal{C}^*}$. Second, while there were errors in derived predictors (e.g., baseline CD4, baseline VL) in the unvalidated dataset, these errors were due to errors in the date of ART initiation and there were no errors in the recorded predictors ($S_X = 0$). Thus, we also did not need to model $\overline{\mathcal{X}}$ as it was perfectly predicted by $\overline{\mathcal{X}^*}$. Therefore, these simplifications allowed us to model

$$f(\overline{\mathcal{A}}, \overline{\mathcal{D}} | \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}, \overline{\mathcal{X}^*}) = f(\overline{\mathcal{A}} | \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}, \overline{\mathcal{X}^*})\ f(\overline{\mathcal{D}} | \overline{\mathcal{A}}, \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}, \overline{\mathcal{X}^*}). \tag{3.3}$$

Because we were only interested in the time of first ART initiation, not all subsequent ART changes, we were able to model the time of first ART initiation directly. Specifically, let $\mathcal{A}^1_m = max_{k \leq m}(\mathcal{A}_k)$ be the indicator that ART had been initiated prior to or during month m. Instead of $f(\overline{\mathcal{A}} | \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}, \overline{\mathcal{X}^*})$, we used $f(\overline{\mathcal{A}^1} | \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}, \overline{\mathcal{X}^*})$ as our model of ART status. This can be further simplified to

$$f(\overline{\mathcal{A}^1} | \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}, \overline{\mathcal{X}^*}) = f(\mathcal{A}^1_0 | \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}, \overline{\mathcal{X}^*}) \prod_{m=1}^{M_{post}} f(\mathcal{A}^1_m | \mathcal{A}^1_{m-1}, \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}, \overline{\mathcal{X}^*}), \tag{3.4}$$

where $Pr(\mathcal{A}^1_m = 1 | \mathcal{A}^1_{m-1} = 1, \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}, \overline{\mathcal{X}^*}) = 1$. Note that in this model we conditioned on $\overline{\mathcal{A}^*}$, the unvalidated vector of new ART drug indicators, rather than $\overline{\mathcal{A}^{1*}} = \{\mathcal{A}^{1*}_0, \mathcal{A}^{1*}_1, ..., \mathcal{A}^{1*}_{M_{post}}\}$, the unvalidated vector of the indicator of having initiated ART, because $\overline{\mathcal{A}^*}$ is richer than $\overline{\mathcal{A}^{1*}}$ and may improve modeling (e.g., if the first date of ART initiation in the unvalidated data is incorrect, the second date of ART initiation in the unvalidated data might be a good candidate for the true first date

of ART initiation). This model was fit using a pooled logistic regression model.

Although we were similarly interested in the first date of ADE, we chose to model all ADEs (i.e., the complete vector $\overline{\mathcal{D}}$), rather than just focusing on the first. Unlike ART status, the variables associated with a given ADE were not likely to differ based on the ordering of the ADE. Because ADE at a specific time is a binary variable, logistic regression was again used for model fitting.

Many VCCC patient records included data for months prior to enrollment; for example, dates of ART use prior to enrollment may have been included in the patient record. It was important to include this information in the analysis (e.g., a patient starting ART prior to enrollment does not meet analysis eligibility criteria). Thus, the time-discretized variables included time prior to enrollment, e.g., $\overline{\mathcal{A}} = \{\mathcal{A}_{M_{pre}}, ..., \mathcal{A}_{-1}, \mathcal{A}_0, \mathcal{A}_1, ..., \mathcal{A}_{M_{post}}\}$ where M_{pre} designated the longest length of pre-enrollment follow-up among all patients.

A total of 32 covariates, $\overline{\mathcal{X}^*}$, were used for the imputation models based on their clinical relevance and a priori belief that they might be predictive of validated values. Time-invariant covariates (calendar year of enrollment, age at first visit, and sex) were attributed to each person-month observation. Time-varying covariates included months since enrollment, current CD4, previous and next CD4, current viral load, previous and next viral loads. Some patients were missing values for certain variables. For laboratory measurements, we carried forward values from months where previous measurements were available. If there were no previous measurements to carry forward, we included an indicator variable denoting that the value was missing. We note that a select number of variables were only considered relevant predictors for one of the two models. For the ART status model, additional time-varying covariates included the unvalidated number of previous ART drug initiations up to that month and during that month. Since the type of ADE in the unvalidated dataset was highly predictive of the presence or absence of an ADE in the validated data, we included dummy variables corresponding to fourteen specific ADEs in the unvalidated data for the ADE status model. All continuous variables were modeled using restricted cubic splines. Twenty iterations were used for the MI procedure. As a sensitivity analysis, we also implemented our TDMI method using reduced imputation models with just four predictor variables: sex, age at first visit, months since enrollment, and current CD4.

There are several possible strategies for undiscretizing data, including an assignment to the start of the one month interval, the end of the interval, the midpoint of the interval, or a random timepoint within the interval. This decision is important

because it may impact the calculated time to event, the assignment of baseline covariates, or even eligibility in the analysis. In our study, we undiscretized values to the start of the interval due to practical considerations: most ART initiations occurred on the same day as enrollment and we wanted imputed ART initiations during that monthly interval to map back to day 0, rather than say day 15 or 30.

For simplicity, we did not adjust for any other variables when modeling the association between baseline CD4 and time to ADE using Cox regression. In practice, one could adjust for relevant confounders (i.e, sex, age, baseline VL, etc.) if desired and appropriate.

3.4 Results

For this section we applied our TDMI procedure to data from the VCCC and compared its performance with estimates obtained via alternative strategies. Specifically, we selected a simple random sample of patient records to serve as our validation subsample. For the records that were not randomly selected, we ignored the validation data (i.e., we pretended that no validation data was available). We then applied our TDMI approach using the unvalidated data on all records together with the validation subsample. TDMI estimates were compared to the naive pre-audit estimates using standard methods on the unvalidated data, the post-audit estimates using standard methods on the fully validated data, and the complete-case estimates using standard methods for only the subset of validated records.

In Figure 3.1, we show the estimated cumulative incidence of ADE over time using the unvalidated dataset, the fully validated dataset, the complete-case analysis using a simple random sample of 1000 validated records, and the TDMI strategy using the same 1000 validated records. For this particular subset, the TDMI estimates appeared to have smaller bias as well as narrower confidence intervals compared to the complete-case estimates. Both the complete-case and the TDMI estimates were closer to the gold standard estimates than the naive estimates at most time points. Specifically, the TDMI estimate of the incidence of ADE at 5 years was 8.9% (95% CI: 7.0% - 10.8%) and the complete-case estimate was 8.2% (95% CI: 4.8% - 11.6%), compared to the naive estimate of incidence at 5 years of 19.6% (95% CI: 17.1% - 22.1%) and the gold standard estimate of 8.3% (95% CI: 6.6% - 10.0%). Similarly, the estimated HR for the association between a 100 cell/mm^3 increase in CD4 count and ADE was 0.66 (95% CI: 0.57 - 0.75) for the TDMI approach and 0.70 (95% CI: 0.56 - 0.89) for the complete-case approach compared to the naive estimate of 0.80

(95%CI: 0.74 - 0.86) and the gold standard estimate of 0.63 (95%CI: 0.55 - 0.72).

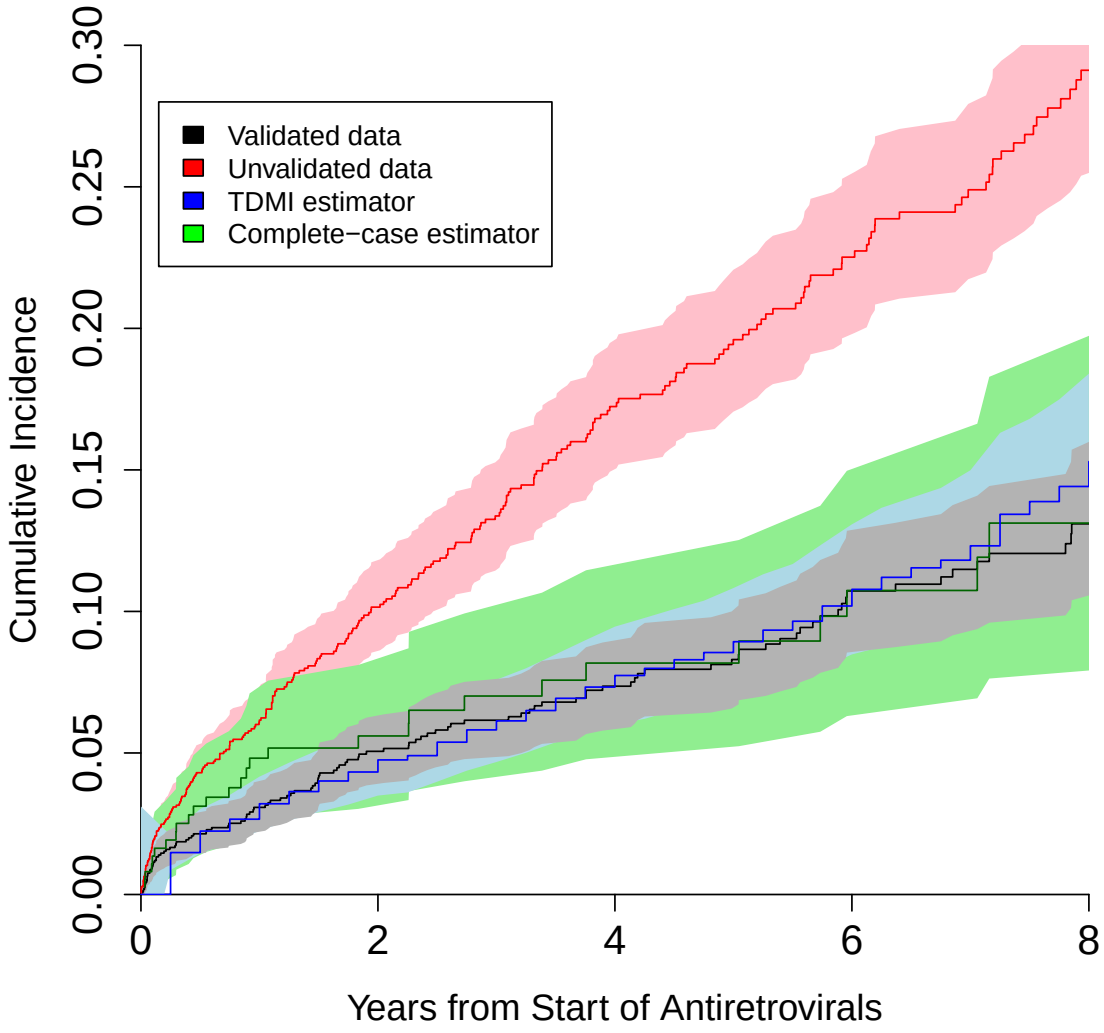


Figure 3.1: Estimated incidence of AIDS-defining event over time using unvalidated, validated, time-discretized modeling and imputation (TDMI), and complete-case approaches. Estimates for TDMI and complete-case approaches are based on one randomly selected iteration.

While promising, these results were based on a single validation sample. Because we had the fully validated data on all patient records, we were able to repeat the process many times, compare estimates to the fully validated data, and empirically study the performance of our TDMI approach. To quantitatively compare approaches, we calculated the difference and the squared difference between each candidate estimate of the 5-year incidence and the log HR to their corresponding estimates based on the complete validated data (i.e., the gold standard estimates) for 1000 replications. The

mean difference (bias), variance, and mean squared difference (mean squared error; MSE) for each candidate estimator (TDMI, complete-case, and naive) were then calculated. In general, the strategy with the lowest MSE was considered the preferred strategy.

Using an audit size of 1000, the MSE of the TDMI estimator for ADE incidence at 5 years was similar, but slightly lower than that of the complete-case estimator (1.5×10^{-4} vs. 2.3×10^{-4}). This result was driven by the TDMI estimator's lower variance (5.8×10^{-5} vs. 2.3×10^{-4}), despite a larger absolute bias (-0.0098 vs. 0.0007). The MSE for the TDMI estimator for the log HR was substantially lower than that of the complete-case estimator (0.003 vs. 0.021). The bias and variance of the TDMI estimator for the log HR were 0.027 and 0.002, respectively, compared with -0.020 and 0.020 for the complete-case estimator.

Our original selection of an audit size of 1000 records was chosen *a priori* but was arbitrary. To assess how the TDMI approach performed for varying audit sizes, we repeated the entire exercise for various audit sample sizes, ranging from $n = 100$ to $n = 4000$ records. The MSEs for the estimated log HR and incidence at 5 years across various audit sizes are shown in Figure 3.2, as well as a bias-variance decomposition of MSEs. With validation sample sizes of 400 or greater, both the TDMI estimators and the complete-case estimators had lower MSE than the naive estimators. For the log HR, the TDMI estimator beat the naive estimator's MSE with an audit sample size of only 100 and had lower MSE than the complete-case estimator at all audit sample sizes. When estimating the incidence of ADE at 5 years, the MSE for the TDMI approach tended to be higher than the complete-case analysis for audit sample sizes less than 500, but fairly similar thereafter. In general, the TDMI estimator was less variable but more biased, particularly at the smaller audit sizes, than the complete-case estimator.

Figure 3.2 also includes results using the TDMI approach with a reduced imputation model that included fewer predictor variables. TDMI estimates of the log HR using the reduced imputation models were similar to the original TDMI estimates, with a slight improvement in the MSE when the audit sample size was less than 500 but higher otherwise; the MSEs remained lower relative to the complete case estimator at all subset sample sizes. In contrast, the TDMI approach based on these reduced imputation models performed worse when estimating the incidence of ADE at 5 years. The bias of these new estimates was such that the TDMI estimator had a higher MSE than the complete case estimator at all audit sample sizes. These results are intuitive and highlight the challenges of model fitting at varying audit sizes. The

original model was chosen for an audit size of 1000 records; with small audit sizes, fitting similarly complicated models can lead to over-fitting and resulting bias, as seen with the log HR. In contrast, in the reduced model we did not include specific types of ADEs, which were very predictive of having any ADE; therefore, the estimated incidence of ADE from the reduced model was more biased, likely due to poor model specification.

3.5 Simulation

We conducted a simulation study to better understand how the TDMI approach performs when the imputation model is misspecified. Simulated data were based on a simplified version of the VCCC example where we are interested in the association between a predictor variable and a time-to-event outcome and the incidence of that outcome at a fixed timepoint.

The simulated cohort included 4000 subjects each with 100 months of follow-up. Each subject was assigned two continuous, correlated variables, X_1 and X_2 drawn from a bivariate normal distribution with mean 0, variance 1, and covariance σ. For simplicity, these variables were time-invariant. $\mathcal{A}^*_m$ was drawn from a Bernoulli distribution at month $m = 1, \ldots, 100$ with the logit probability of success equal to $-3-0.02m$. $\mathcal{D}^*_m$ was drawn from a Bernoulli distribution with the logit probability of success equal to $-5-0.02m+0.5\mathcal{A}^*_m$. $\mathcal{A}_m$ was then drawn from a Bernoulli distribution with the logit probability of success equal to $-5-0.02m-X_1+\beta_2 X_2+4\mathcal{A}^*_m+0.5\mathcal{D}^*_m$. Finally, $\mathcal{D}_m$ was drawn from a Bernoulli distribution with the logit probability of success equal to $-7-0.02m-2X_1+\gamma_2 X_2+4\mathcal{D}^*_m+0.5\mathcal{A}_m$. The parameters (β_2, γ_2) were set to (1, 2), (0.5, 1), (0.25, 0.5), and (0, 0) in different simulations to represent varying strengths of association between X_2 and $(\overline{\mathcal{A}}, \overline{\mathcal{D}})$; σ was varied between $-0.25, 0$ and 0.25. The dates of ART initiation, T_0, and ADE, T_E, were computed as the smallest values of m with $\mathcal{A}_m = 1$ and $\mathcal{D}_m = 1$, respectively. If $\mathcal{A}_m = 0$ (or similarly $\mathcal{D}_m = 0$) for all m, then T_0 (similarly T_E) was set to an arbitrary value bigger than 100 (e.g., 101). Records were eligible for analysis if $W = I(T_0 \leq 100)I(T_0 < T_E) = 1$. Then $Y = min(T_E, 100) - T_0$ and $D = I(T_E \leq 100)$. W^*, Y^*, and D^* were similarly computed using $\overline{\mathcal{A}^*}$ and $\overline{\mathcal{D}^*}$. The parameters of interest were, among those with $W = 1$, $\rho = P(T_E - T_0 \leq 60)$ and β from the proportional hazards model, $\lambda(m|X_1) = \lambda_0(m)exp(\beta X_1)$.

A subset of 1000 subjects were randomly selected to represent an audited cohort with $\overline{\mathcal{A}}$ and $\overline{\mathcal{D}}$ (and hence, W, Y, and D) known; for the remaining 3000 subjects, $\overline{\mathcal{A}}$

Table 3.2: Summary of simulation results for time-discretized modeling and imputation (TDMI) parameter estimates from Cox regression with different levels of misspecification in the imputation model.

Fixed values		Cox regression parameter estimate							
(β_2,γ_2)	σ	Truth	Naive	Fully specified			Misspecified		
				Bias	MSE	Coverage	Bias	MSE	Coverage
(1,2)	−0.25	−1.14	0.00	0.007	0.0033	0.94	−0.402	0.1761	0.06
(1,2)	0	−0.90	−0.01	−0.003	0.0027	0.94	−0.415	0.1859	0.03
(1,2)	0.25	−0.73	0.00	−0.001	0.0026	0.95	−0.386	0.1607	0.04
(0.5,1)	−0.25	−1.71	0.00	−0.005	0.0063	0.93	−0.322	0.1183	0.19
(0.5,1)	0	−1.52	−0.00	−0.008	0.0060	0.94	−0.311	0.1119	0.21
(0.5,1)	0.25	−1.38	−0.00	−0.010	0.0071	0.93	−0.267	0.0859	0.33
(0.25,0.5)	−0.25	−2.00	−0.01	−0.004	0.0082	0.93	−0.120	0.0265	0.74
(0.25,0.5)	0	−1.86	−0.00	−0.019	0.0084	0.93	−0.133	0.0287	0.72
(0.25,0.5)	0.25	−1.76	0.00	−0.012	0.0088	0.94	−0.112	0.0239	0.76
(0,0)	−0.25	−2.05	−0.00	0.002	0.0085	0.93	−0.003	0.0082	0.94
(0,0)	0	−2.08	0.01	0.022	0.0088	0.94	0.017	0.0085	0.94
(0,0)	0.25	−2.07	−0.00	0.014	0.0092	0.93	0.014	0.0091	0.93

Truth and *Naive* refer to asymptotic estimates of the parameter values using only validated or unvalidated records, respectively (n=500,000).

and $\overline{\mathcal{D}}$ (and therefore W, Y, and D) were treated as missing. $\overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, X_1$, and X_2 were treated as known for all 4000 subjects. The TDMI procedure was implemented to multiply impute missing values of $\overline{\mathcal{A}}$ and $\overline{\mathcal{D}}$ and then to derive (W, Y, D) for the 3000 subjects. Two candidate sets of imputation models were considered for the TDMI procedure: (i) perfectly specified models for $\overline{\mathcal{A}}$ and $\overline{\mathcal{D}}$ that included X_1 and X_2, and (ii) misspecified models that did not include X_2. The parameters of interest were estimated using Kaplan-Meier estimates and Cox regression applied to the multiply imputed data. A total of 12 scenarios were constructed by varying the three undefined parameters $(\sigma, \beta_2, \gamma_2)$ to allow varying amounts of potential misspecification. For each scenario, estimates and corresponding 95% confidence intervals for both parameters from both the perfectly specified and misspecified TDMI implementations were generated for 1000 independent simulations.

Tables 3.2 and 3.3 show the bias, MSE, and coverage of the TDMI approach under different simulation settings. When the imputation model was correctly specified, estimates of the 60-month incidence and log HR were approximately unbiased with coverage probabilities at or just below the nominal level (93% - 95%) as expected. When the imputation model was incorrectly specified, absolute bias increased and coverage decreased as the relative strength of association for the omitted covariate increased.

Table 3.3: Summary of simulation results for time-discretized modeling and imputation (TDMI) parameter estimates from Kaplan-Meier estimation with different levels of misspecification in the imputation model.

Fixed values		Kaplan-Meier estimate for $P(T_E - T_0 \leq 60)$							
(β_2,γ_2)	σ	Truth	Naive	Fully specified			Misspecified		
				Bias	MSE	Coverage	Bias	MSE	Coverage
(1,2)	−0.25	0.77	0.84	−0.000	0.0001	0.94	−0.090	0.0084	0.00
(1,2)	0	0.79	0.84	0.001	0.0001	0.93	−0.080	0.0067	0.01
(1,2)	0.25	0.82	0.84	0.001	0.0001	0.94	−0.053	0.0031	0.13
(0.5,1)	−0.25	0.82	0.84	−0.002	0.0001	0.95	−0.016	0.0004	0.75
(0.5,1)	0	0.84	0.84	−0.000	0.0001	0.94	−0.012	0.0003	0.83
(0.5,1)	0.25	0.86	0.84	−0.000	0.0001	0.95	−0.006	0.0001	0.92
(0.25,0.5)	−0.25	0.84	0.84	−0.000	0.0001	0.95	−0.003	0.0001	0.93
(0.25,0.5)	0	0.85	0.84	0.001	0.0001	0.94	−0.001	0.0001	0.95
(0.25,0.5)	0.25	0.87	0.84	−0.000	0.0001	0.95	−0.002	0.0001	0.94
(0,0)	−0.25	0.86	0.84	0.001	0.0001	0.94	0.000	0.0001	0.95
(0,0)	0	0.86	0.84	0.001	0.0001	0.93	0.001	0.0001	0.94
(0,0)	0.25	0.86	0.84	−0.001	0.0001	0.93	−0.001	0.0001	0.93

Truth and *Naive* refer to asymptotic estimates of the parameter values using only validated or unvalidated records, respectively (n=500,000).

3.6 Discussion

Using EHR data from an HIV cohort, we have illustrated the bias that can arise by ignoring data errors, and we have proposed a missing data analysis solution that incorporates validation data to address multidimensional errors in time-to-event analyses. To our knowledge, this is the first study to simultaneously address errors in both predictors and outcomes in a time-to-event analysis. We were also able to address errors in study eligibility. The TDMI approach did not outperform the complete-case approach under all scenarios, but we are encouraged that it led to improved estimation under certain conditions, particularly when estimating the log HR.

The TDMI procedure is subject to various assumptions generally similar to those required for multiple imputation in standard missing data settings (Schafer, 1999). The key missing at random assumption was easily satisfied in our example as the audited sample was a simple random sample. This assumption can also be satisfied in more complicated settings where subjects are sampled with known probabilities, but will likely be violated if the validation sample is one of convenience.

Another basic assumption underlying the TDMI approach is that the imputation model is properly specified. This requires the identification of covariates in the unvalidated dataset predictive of values in the validated dataset as well as a model that properly specifies the relationships. This is difficult in practice. Despite our best

efforts – the incorporation of over 30 covariates, both time-fixed and time-varying exposures – estimates for our approach were still biased, especially at smaller validation sample sizes. Results from both our reduced model TDMI and the simulation study highlight potential challenges with model misspecification. Model overfitting can be a problem at smaller validation sample sizes, as seen by our reduced model TDMI out-performing the original model TDMI designed for an audit size of 1000 when estimating the log HR. But the reduced model was not sufficiently rich to obtain good estimates for the incidence of ADE at modest audit sizes.

We estimated standard errors using the Robins and Wang (2000) imputation variance estimator instead of the more popular (and easier to implement) approach proposed by Rubin (Little and Rubin, 2014), because of incompatibility between imputation and analysis models. There were two sources of incompatibility in our setting. First, the unit of observation was different between the imputation model (subject-month observations) and the analysis model (subject-level observations). Second, our study had exclusion criteria that removed observations from the analysis model that contributed information to the imputation model. Standard errors calculated using Rubin's rule in our setting led to inflated standard errors and conservative confidence intervals (e.g., coverage of 98-99% in simulations, data not shown).

We considered alternative modeling approaches (e.g., classification and regression trees, random forests, support vector machines, and linear discriminant analysis), but ultimately fit logistic regression models. Given the improbability of knowing, *a priori*, which model will perform best for a certain setting, it might be worthwhile to add a preliminary step that selects the most appropriate model through cross-validation or some other model-selection procedure in the audit subsample. There is certainly a bias versus variance trade-off with using the TDMI approach. The complete-case analysis is unbiased but generally more variable than TDMI. We are currently studying raking methods to combine potentially biased but efficient estimators like the TDMI estimator with unbiased but less efficient complete-case estimators (Lumley et al., 2011).

The size of the validation subsample is clearly important for the performance of our approach. In this study, a validation subsample of approximately 200-300 records was needed to obtain estimates of the 5-year incidence of ADE with lower MSE than the naive analysis on unvalidated data. We suspect that the high MSE of our estimators of the incidence at smaller sample sizes was due in part to the sparse occurrences of ADEs and their corresponding errors in the audited subset. In contrast, the TDMI estimator of the log HR had lower MSE than the naive estimator

with an audit sample size of only 100 records.

There is a potential loss of information from coarsening the data into time intervals when fitting the imputation models. Once those values have been imputed, the analyst must also convert the data back to the original unit of measurement. Losses of information due to discretization to the level of months will be minimal in clinical settings where visits typically occur no more than monthly. For example, as a sensitivity analysis, we also coarsened the data from the validated records into monthly intervals and re-estimated the incidence of ADE at 5 years for a validation subsample of size 1000. The MSE of this discretized version of the complete-case estimator (1.7×10^{-4}) was similar to both the non-discretized TDMI estimator (1.5×10^{-4}) and the complete-case estimator (2.3×10^{-4}).

We acknowledge that, for the notation in Section 3.3.1, we had two variables corresponding to an error term: an indicator of the existence of an error and the magnitude of an error. However, in our implementation, we assume that the errors arose from the truth plus some error. The distinction is subtle, but this means we did not first model for the presence of any error when imputing values based on the unvalidated data. This decision was made for practical purposes - to reduce the number of imputation models from four to two - but may have resulted in additional noise to values that were error-free in the unvalidated dataset.

Although our analyses focused on Kaplan-Meier and Cox regression estimates, a strength of our multiple imputation approach is that we could have also performed other estimation procedures. This is important because analyses of EHR data typically require addressing multiple problems simultaneously (e.g., confounding, missing data, and informative censoring). Methods for dealing with these other sources of bias could potentially be applied to the multiply imputed dataset without substantial modification. Of course, the performance of our approach may vary across analysis methods, as we saw in this study.

Future research will consider improving the efficiency of these methods by applying principles of two-phase designs, such as oversampling exposures or events that are rare or considered *a priori* to be more error-prone.

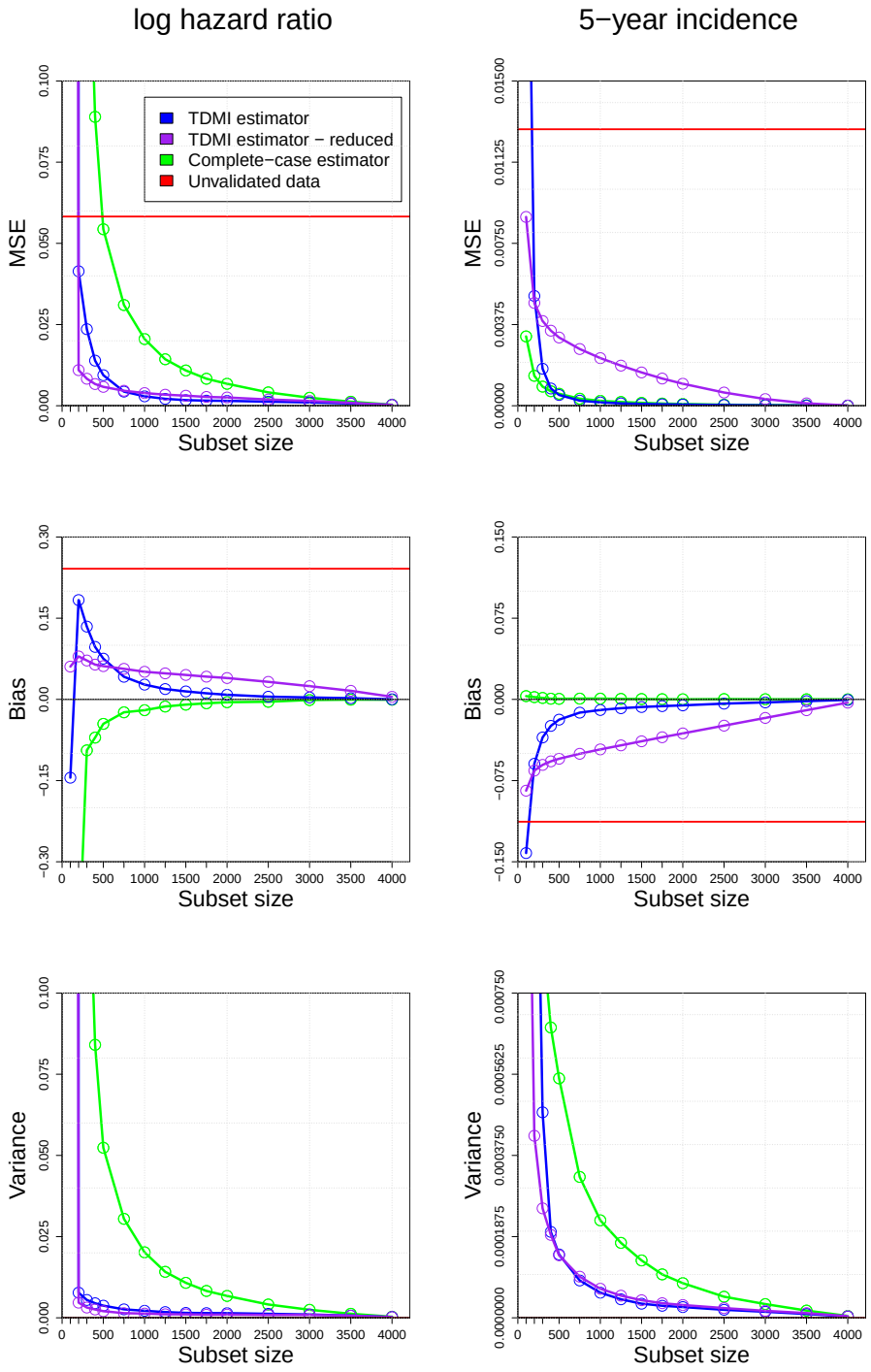


Figure 3.2: Mean square errors (top row), bias (middle row), and variance (bottom row) for estimates of the log hazard ratio (first column) and 5-year incidence of AIDS-defining event (second column) from each candidate estimator (*including a TDMI estimator where fewer predictor variables are included in the imputation model*) and various audit sizes. Estimates are calculated as the average of 1000 replications.

3.7 Appendix B

3.7.1 Details of model specification

In this section, we provide additional details regarding our modeling procedure. Recall, we fit two logistic regression models to model ART status and the occurrence of ADEs each month in the validated dataset (Web Table 1).

In this dataset, the longest length of post-enrollment follow-up (M_{post}) was 167 months. The maximum number of months prior to enrollment was set at 100. For each patient, only data from months prior to their month of last follow-up was included. To assign dates into monthly intervals for a given patient, we subtracted the date of enrollment (T_B), divided by 30.437 to convert days into months, and rounded down to the nearest integer. For example, the length of follow-up after enrollment (in months) was calculated as $\left\lfloor \frac{max(T_V)-T_B}{30.437} \right\rfloor$, where $\lfloor . \rfloor$ denotes rounding down to the nearest integer and T_V corresponds to patients' visit dates. Therefore, $\mathcal{C}_m = 1$ if $m = \left\lfloor \frac{max(T_V)-T_B}{30.437} \right\rfloor$ and $\mathcal{C}_m = 0$ otherwise. The variables $\mathcal{A}_m, \mathcal{D}_m$, and $\mathcal{X}_m$, and their unvalidated versions, were calculated similarly.

Each logistic regression model was fit separately using different covariates. Restricted cubic splines with three knots were used for all continuous variables; knot locations were set using the defaults of the Design package in R. The specification of the linear predictors for each of the two models is provided on the next page (Models 1 and 2). For simplicity, we simply list the linear terms (i.e., the splines are implicit). If a model could not be fit (due to small sample size), we re-ran without splines and started removing variables one at a time; previous and future lab measurements as well as specific ADEs with low frequencies were removed first. We continued to remove variables until the logistic regression model converged or we were left with an intercept-only model. All statistical analyses were performed using R version 3.4.1. For the models corresponding to ART status (Model 1), observations occurring after the month of ART initiation in the validated dataset $\left(m > \left\lfloor \frac{min(T_A)-T_B}{30.4} \right\rfloor \right)$ were excluded.

A description of all variables used in each of the two models is provided in 3.5. All variables from the unvalidated dataset are denoted using an asterisk. For variables where a value may be missing (e.g., a patient may not a documented CD4 count prior to a given month), we recorded a value of 0 and added an indicator variable for whether the value was observed. Many variables were included in both fitted models, including the current month and the following from the unvalidated dataset:

new ART drug initiation in a given month, ADE in a given month, sex, year of enrollment, and CD4/VL measurements at different time points.

The model corresponding to ART status (Models 1) included additional indicator variables from the invalidated dataset corresponding to a new drug initiation in the previous month, next month, or at any time during follow-up. For the model corresponding to ADE events (Model 2), indicator variables from the unvalidated dataset corresponding to specific ADEs (cytomegalovirus, encephalopathy/dementia, candidiasis, histoplasmosis, Kaposi's sarcoma, lymphoma, mycobacterium avian complex, pneumocystis pneumonia, pneumonia, retinitis, tuberculosis, wasting, weight loss, and other) as well as any ADE recorded prior to month m were included; additionally, an indicator variable from the validated dataset corresponding to a new ART drug initiation the current month was also included. An additional indicator term is included in Models 1 and 2 for the month of enrollment, as splines did not provide the flexibility to model the high occurrence of new ART drug initiations or ADE events during month of enrollment.

Table 3.4: Models

Model 1: ART status	$f(\mathcal{A}_m \vert \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}, \overline{\mathcal{X}^*}, max_{k<m}(A_k) = 0) = logit^{-1}(LP_{ART})$
Model 2: ADE	$f(\mathcal{D}_m \vert \mathcal{A}_m, \overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, \overline{\mathcal{C}^*}, \overline{\mathcal{X}^*}) = logit^{-1}(LP_{ADE})$

Model 1: ART status

$$
\begin{aligned}
LP_{ART} =& \beta_0 + \beta_1 \mathcal{A}^*_m + \beta_2 \mathcal{A}^*_{m-1} + \beta_3 \mathcal{A}^*_{m+1} + \beta_4 \max_{k<M}(\mathcal{A}^*_k) + \beta_5 \mathcal{D}^*_m + \\
& \beta_6 \mathcal{X}^*_{Sex} + \beta_7 \mathcal{X}^*_{EnrollYear} + \beta_8 \mathcal{X}^*_{EnrollAge} + \\
& \beta_9 \mathcal{X}^*_{CD4,m} + \beta_{10} \mathcal{X}^*_{CD4.time,m} + \beta_{11} \mathcal{X}^*_{CD4.missing,m} + \\
& \beta_{12} \mathcal{X}^*_{VL,m} + \beta_{13} \mathcal{X}^*_{VL.time,m} + \beta_{14} \mathcal{X}^*_{VL.missing,m} + \\
& \beta_{15} \mathcal{X}^*_{CD4,m-1} + \beta_{16} \mathcal{X}^*_{CD4.missing,m-1} + \beta_{17} \mathcal{X}^*_{VL,m-1} + \beta_{18} \mathcal{X}^*_{VL.missing,m-1} + \\
& \beta_{19} \mathcal{X}^*_{Next.CD4,m} + \beta_{20} \mathcal{X}^*_{Next.CD4.missing,m} + \\
& \beta_{21} \mathcal{X}^*_{Next.VL,m} + \beta_{22} \mathcal{X}^*_{Next.VL.missing,m} + \\
& \beta_{23} m + \beta_{24} mzero
\end{aligned}
$$

Model 2: ADE

$$
\begin{aligned}
LP_{ADE} =& \delta_0 + \delta_1 \mathcal{A}_m + \delta_2 \mathcal{A}^*_m + \delta_3 \mathcal{D}^*_m + \delta_4 \max_{k<m}(\mathcal{D}^*_k) + \delta_5 \mathcal{X}_{ADE.NODxDate} + \\
& \delta_6 \mathcal{X}^*_{Sex} + \delta_7 \mathcal{X}^*_{EnrollYear} + \delta_8 \mathcal{X}^*_{EnrollAge} + \\
& \delta_9 \mathcal{X}^*_{CD4,m} + \delta_{10} \mathcal{X}^*_{CD4.time,m} + \delta_{11} \mathcal{X}^*_{CD4.missing,m} + \\
& \delta_{12} \mathcal{X}^*_{VL,m} + \delta_{13} \mathcal{X}^*_{VL.time,m} + \delta_{14} \mathcal{X}^*_{VL.missing,m} + \\
& \delta_{15} \mathcal{X}^*_{CD4,m-1} + \delta_{16} \mathcal{X}^*_{CD4.missing,m-1} + \delta_{17} \mathcal{X}^*_{VL,m-1} + \delta_{18} \mathcal{X}^*_{VL.missing,m-1} + \\
& \delta_{19} \mathcal{X}^*_{Next.CD4,m} + \delta_{20} \mathcal{X}^*_{Next.CD4.missing,m} + \\
& \delta_{21} \mathcal{X}^*_{Next.VL,m} + \delta_{22} \mathcal{X}^*_{Next.VL.missing,m} + \\
& \delta_{23} \mathcal{X}^*_{ADE.Candid,m} + \delta_{24} \mathcal{X}^*_{ADE.Cyto,m} + \delta_{25} \mathcal{X}^*_{ADE.Enceph,m} + \\
& \delta_{26} \mathcal{X}^*_{ADE.Histo,m} + \delta_{27} \mathcal{X}^*_{ADE.KS,m} + \delta_{28} \mathcal{X}^*_{ADE.Lymph,m} + \\
& \delta_{29} \mathcal{X}^*_{ADE.MAC,m} + \delta_{30} \mathcal{X}^*_{ADE.Meningitis,m} + \delta_{31} \mathcal{X}^*_{ADE.PCP,m} + \\
& \delta_{32} \mathcal{X}^*_{ADE.Pneumonia,m} + \delta_{33} \mathcal{X}^*_{ADE.Retinitis,m} + \delta_{34} \mathcal{X}^*_{ADE.TB,m} + \\
& \delta_{35} \mathcal{X}^*_{ADE.Wasting,m} + \delta_{36} \mathcal{X}^*_{ADE.Weight,m} + \delta_{37} \mathcal{X}^*_{ADE.Other,m} + \\
& \delta_{38} m + \delta_{39} mzero
\end{aligned}
$$

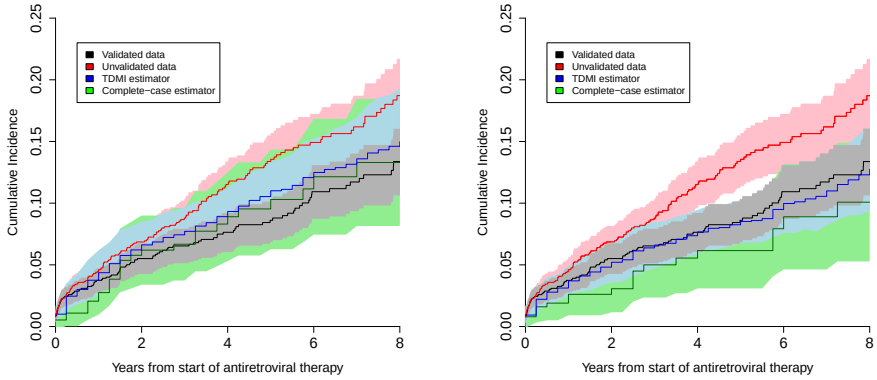
Table 3.5: Description of variables included in prediction models.

Variable	Description	Model 1	Model 2
m	current month	Yes	Yes
$mzero$	indicator for month 0	Yes	Yes
$\mathcal{A}^*_m$	new ART drug initiation in the current month (yes/no)	Yes	Yes
$\mathcal{A}^*_{m-1}$	new ART drug initiation in the previous month (yes/no)	Yes	No
$\mathcal{A}^*_{m+1}$	new ART drug initiation in the next month (yes/no)	Yes	No
$\max_{k<M}(\mathcal{A}^*_k)$	an indicator for any ART initiation at any time	Yes	No
$\mathcal{D}^*_m$	documented ADE in month m (yes/no)	Yes	Yes
$\max_{k<m}(\mathcal{D}^*_k)$	any ADE recorded prior to month m (yes/no)	No	Yes
$\mathcal{X}^*_{Sex}$	sex	Yes	Yes
$\mathcal{X}^*_{EnrollYear}$	year of enrollment	Yes	Yes
$\mathcal{X}^*_{CD4,m}$	CD4 count at month m	Yes	Yes
$\mathcal{X}^*_{CD4.missing,m}$	an indicator of no CD4 count prior/during month m	Yes	Yes
$\mathcal{X}^*_{CD4.time,m}$	months since CD4 count was last measured	Yes	Yes
$\mathcal{X}^*_{CD4,m-1}$	CD4 count at month $m-1$	Yes	Yes
$\mathcal{X}^*_{CD4.missing,m-1}$	an indicator of no CD4 count prior/during month $m-1$	Yes	Yes
$\mathcal{X}^*_{Next.CD4,m}$	next CD4 count after month m	Yes	Yes
$\mathcal{X}^*_{Next.CD4.missing,m}$	an indicator of no CD4 count after month m	Yes	Yes
$\mathcal{X}^*_{VL,m}$	log viral load measurement at month m	Yes	Yes
$\mathcal{X}^*_{VL.missing,m}$	log viral load measurement prior/during month m	Yes	Yes
$\mathcal{X}^*_{VL.time,m}$	months since log viral load was last measured	Yes	Yes
$\mathcal{X}^*_{VL,m-1}$	log viral load at month $m-1$	Yes	Yes
$\mathcal{X}^*_{VL.missing,m-1}$	an indicator of no log viral load prior/during month $m-1$	Yes	Yes
$\mathcal{X}^*_{Next.VL,m}$	next log viral load after month m	Yes	Yes
$\mathcal{X}^*_{Next.VL.missing,m}$	an indicator of no log viral load after month m	Yes	Yes
$\mathcal{X}^*_{ADE.NODxDate}$	any ADE recorded without a specific diagnosis date (yes/no)	No	Yes
$\mathcal{X}^*_{ADE.Cyto,m}$	indicator variable for cytomegalovirus	No	Yes
$\mathcal{X}^*_{ADE.Enceph,m}$	indicator variable for encephalopathy/dementia	No	Yes
$\mathcal{X}^*_{ADE.Candid,m}$	indicator variable for candidiasis	No	Yes
$\mathcal{X}^*_{ADE.Histo,m}$	indicator variable for histoplasmosis	No	Yes
$\mathcal{X}^*_{ADE.KS,m}$	indicator variable for Kaposi's sarcoma	No	Yes
$\mathcal{X}^*_{ADE.Lymph,m}$	indicator variable for lymphoma	No	Yes
$\mathcal{X}^*_{ADE.MAC,m}$	indicator variable for mycobacterium avian complex	No	Yes
$\mathcal{X}^*_{ADE.Mengitis,m}$	indicator variable for meningitis	No	Yes
$\mathcal{X}^*_{ADE.PCP,m}$	indicator variable for pneumocystis pneumonia	No	Yes
$\mathcal{X}^*_{ADE.Pneumonia,m}$	indicator variable for pneumonia	No	Yes
$\mathcal{X}^*_{ADE.Retinitis,m}$	indicator variable for retinitis	No	Yes
$\mathcal{X}^*_{ADE.TB,m}$	indicator variable for tuberculosis	No	Yes
$\mathcal{X}^*_{ADE.Wasting,m}$	indicator variable for wasting	No	Yes
$\mathcal{X}^*_{ADE.Weight,m}$	indicator variable for weight loss	No	Yes
$\mathcal{X}^*_{ADE.Other,m}$	indicator variable for other	No	Yes
$\mathcal{A}_m$	new ART initiation in the current month (validated dataset)	No	Yes

Abbreviations: ADE, AIDS-defining event. ART, antiretroviral therapy.

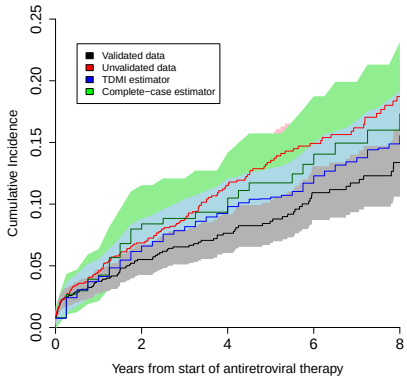
3.7.2 Additional figures

The following plots illustrate concepts discussed in the original manuscript, but not included due to space constraints. Figure 3.3 is an extension of Figure 3.1 in the main text, plotting the estimated incidence of AIDS-defining events over time for six replications.

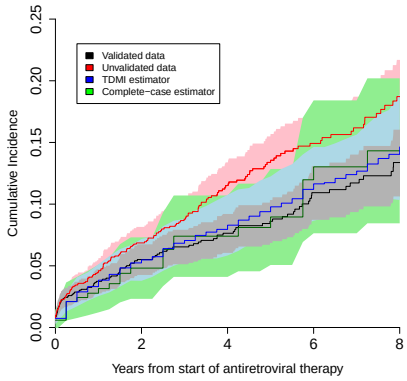


Replication 1

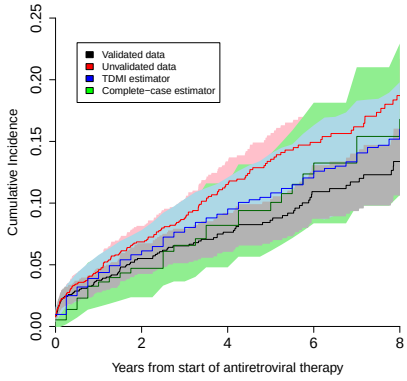
Replication 2



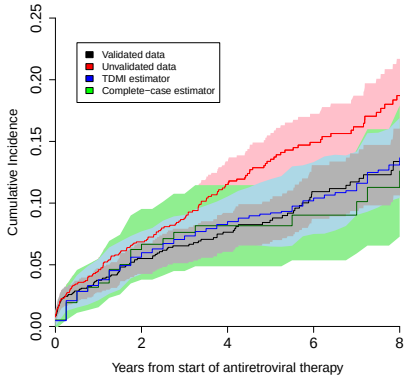
Replication 3



Replication 4

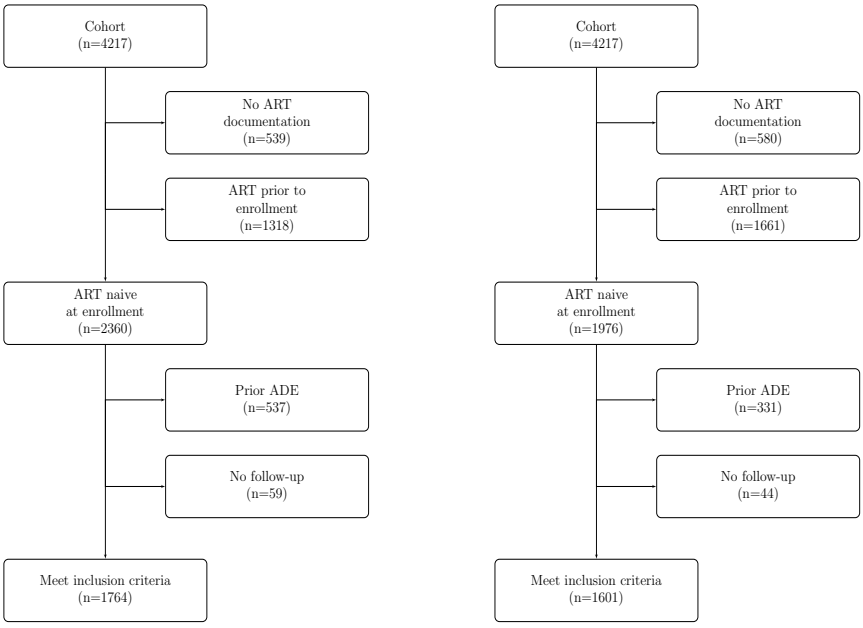


Replication 5



Replication 6

Figure 3.3: Estimated incidence of AIDS-defining events over time using unvalidated, validated, time-discretized modeling and imputation (TDMI), and complete-case approaches for six replications with a subsample of size 1000.



(a) Profile of the unvalidated dataset (N=4217) (b) Profile of the validated dataset (N=4217)

Figure 3.4: Cohort profiles for unvalidated and validated datasets.

CHAPTER 4

A TUTORIAL FOR IMPLEMENTING THE MULTIPLE IMPUTATION VARIANCE ESTIMATOR PROPOSED BY ROBINS AND WANG WITH EXAMPLES AND R CODE

4.1 Introduction

Multiple imputation is a popular statistical method used to account for both missing data and measurement error. With the increasing availability and popularity of statistical software packages containing imputation functions (e.g., *mi*, *aRegImpute*, and *mice* in *R*), researchers of varying statistical backgrounds are now able to incorporate multiple imputation procedures in their analyses. Unfortunately, these analyses are often performed by investigators with limited statistical understanding of how multiple imputation works and the assumptions being made. This unfamiliarity with the procedures being implemented may potentially lead to invalid inferences.

An imputation variance estimator originally proposed by Rubin (Little and Rubin, 2014) has become popular due to its ease of implementation. However, this variance estimator has been shown to be biased when the imputation model is misspecified or if there is incompatibility between the imputation model and the analysis model (Robins and Wang, 2000). Incompatibility can arise when assumptions differ between the analysis and imputation models, or when subjects used in the analysis model are a subset of those used in the imputation model. Incompatibility between imputation and analysis models frequently occurs in many practical analyses and can lead to substantially biased variance estimates.

One example of incompatibility leading to biased variance estimation arises from the use of MI to address errors in variables (Cole et al. (2006); Shepherd et al. (2012); Edwards et al. (2013)). Given that observational data from electronic health records (EHRs) are often incomplete and error-prone, data validation is necessary for correct inference. However, data validation of all patient records is usually not feasible, and instead a random subsample of records may be validated. If the validation subsample is a random sample, then the unobserved validated values are missing at random, and in the validated subsample one can model the association between the validated variables and the unvalidated variables, and from this model multiply impute missing validated values for the unvalidated records. This multiple imputation procedure can result in unbiased estimation if imputation models are properly specified. Analyses using EHR data typically incorporate inclusion/exclusion criteria to determine what

records will be included in analyses. Frequently, imputation is performed prior to record exclusion, or record exclusion is based on imputed values. When this happens, the imputation model is a super-set of the analysis model, the two models are therefore incompatible, and the standard variance formula for MI estimators tends to over-estimate the true variance.

Fortunately, Robins and Wang (2000) (RW) derived an alternative approach for estimating the variance of MI estimators that is able to obtain asymptotically unbiased estimates of the variance in settings with misspecification or incompatibility. While this RW imputation variance is fairly well known among statisticians doing methods research in missing data, it has been rarely implemented and seems to be almost unknown by most analysts. Compared to Rubin's variance estimator, RW is complex and requires additional calculations from both the imputer and analyst. In their original manuscript, Robins and Wang wrote that they "hope that, in the future, software developers will create packages" to implement their approach. Hughes, Sterne, and Tilling (HST) implemented the RW approach for some simple scenarios and showed via simulations that RW out-performed Rubin's rules (RR) with moderate sample sizes (Hughes et al., 2016). Although HST's paper was helpful in clarifying RW, they provided no code for their analyses or simulations. Unfortunately, eighteen years after RW's publication, no existing software packages implement RW, and to our knowledge, there is not even any publicly available code.

The goal of this manuscript is to make this Robins and Wang variance estimator more accessible to future implementors of multiple imputation procedures. We provide several examples of increasing complexity and provide corresponding R code to illustrate how to incorporate RW with different imputation models and analysis models. In the first example, we provide R code to reproduce the RW variance estimate calculation for one of the scenarios in the HST manuscript with linear regression imputation and a subgroup analysis model. The second example includes a joint imputation model for two frequently missing variables, a logistic regression analysis model, and a non-static inclusion criteria (i.e., the post-imputation analysis dataset contains different subjects for each imputation). The third example corresponds to the simulation study performed in an earlier chapter where we implemented our time-discretized modeling and imputation approach and performed analyses using Cox regression and Kaplan-Meier.

4.2 Rubin's variance estimator

Suppose we have a dataset of n subjects with missing values. For the $k = 1, \ldots, m$ imputed datasets, we estimate β^k, our parameter of interest and σ_k^2 the corresponding model variance estimate. The usual imputation variance estimate proposed by Rubin is defined as:

$$\frac{1}{M}\sum_k \hat{\sigma}_k^2 = \frac{1}{M}\sum_k \sigma_k^2 + \left(1 + \frac{1}{M}\right)\frac{1}{M-1}\sum_k (\beta_k - \bar{\beta})^2$$

As mentioned in the introduction, this variance estimator may be biased if there is misspecification in the imputation model or incompatibility between the imputation and analysis models.

The total variance from Rubin's approach can be decomposed into three smaller components: a within-variance component $\left(\frac{1}{M}\sum_k \sigma_k^2\right)$, a component representing the between-variance $\left(\frac{1}{M-1}\sum_k (\beta_k - \bar{\beta})^2\right)$, and an extra simulation variance component $\left(\frac{1}{M*(M-1)}\sum_k (\beta_k - \bar{\beta})^2\right)$.

A closer investigation of these components reveals the limitations of Rubin's approach when there is incompatibility. This is most notable for the the within-variance component. Recall, this component represents a measure of variability caused by the fact that we are taking a sample rather than the entire population. It is calculated based on the sample used in the analysis dataset. However, it does not take into consideration the contributions from the imputation model. Thus, for say a logistic regression model where there is a mean-variance relationship, this component will not change regardless of the size of the dataset used for imputation. This leads to inflated total variance estimates as the incompatibility between imputation and analysis models increases.

4.3 Robins and Wang variance estimator

In the section below, we review how the Robins and Wang imputation variance estimator is calculated. For additional details, we recommend reviewing the original RW manuscript or the HST manuscript.

Suppose we have a dataset of n subjects with missing values. Our p parameter imputation model is fit by estimating θ among the subjects with non-missing values and we generate m imputation datasets. For those $k = 1, \ldots, m$ imputed datasets, we

estimate β^k, our parameter of interest. The Robins and Wang imputation variance estimate Γ is defined as:

$$\Gamma = \frac{1}{n}\tau^{-1}\Delta(\tau^{-1})^T, \qquad \text{where} \tag{4.1}$$

$$\Delta = \Omega + \kappa\Lambda\kappa^T + \frac{1}{n}\{\kappa d_i^T \bar{u_i} + (\kappa d_i^T \bar{u_i})^T\}, \tag{4.2}$$

$$\Omega = \frac{1}{n}\sum_{i=1}^{n} \bar{u_i}^T \bar{u_i}, \tag{4.3}$$

$$\Lambda = \frac{1}{n} d_i^T d_i, \tag{4.4}$$

$$\kappa = \frac{1}{n * m}\sum_{i=1}^{n}\sum_{k=1}^{m}(u_i(\hat{\theta}, \hat{\beta^k}))^T S_i^{mis,k}, \qquad \text{and} \tag{4.5}$$

$$\bar{u_i} = \frac{1}{m}\sum_{k=1}^{m}(u_i(\hat{\theta}, \hat{\beta^k})) \tag{4.6}$$

There are four terms (S^{mis}, d, u, and τ) in the formula above that need to be calculated based on either the imputation model or the analysis model. We briefly describe these components and their calculation below.

4.3.1 Imputation model components

Two of these components (S^{mis}, d) are derived from the imputation model. Both values are based on the score function and its derivative for the imputation model. Let $V = 1$ denote that the observation was observed and $V = 0$ denote that the observation was imputed. For the k^{th} imputation, $S^{mis,k}$ is a $n \times p$ matrix corresponding to the score of each parameter in the imputation model evaluated for each observation that was imputed. However, if the observation was not imputed, a value of 0 is assigned.

$$S_i^{mis,k} = \left[\frac{\partial log f(y_i|X,\theta)}{\partial\theta}|_{\theta=\hat{\theta}}\right] \times I(V_i = 0) \tag{4.7}$$

To calculate d, we first take derivates of the score function with respect to each parameter, evaluate at each observation that was not imputed, and take the average. We then take the inverse of this $p \times p$ matrix, multiply by the transpose of S^{obs}, and multiply by -1. Note that S^{obs} is the score evaluated for each observation that was

not imputed; imputed values are assigned a value of 0.

$$d_i^T = -\left[\frac{1}{n}\sum_{i=1}^{n} I(V_i = 1) \times \frac{\partial}{\partial \theta^T}\left(\frac{\partial log f(y_i|X,\theta)}{\partial \theta}\right)|_{\theta=\hat{\theta}}\right]^{-1} \times S_i^{obs^T} \tag{4.8}$$

4.3.2 Analysis model components

The final two components (u, τ) are derived from the analysis model. Both values are based on the estimating equations pertaining to the analysis model. For each imputation, we evaluate the estimating equation $u_i(\hat{\theta}, \hat{\beta}^k)$ for all subjects in the analysis dataset. To calculate τ, we take the derivative of the estimating equation and evaluate as:

$$\tau = -\frac{1}{nm}\sum_{i=1}^{n}\sum_{k=1}^{m}\left(\frac{\partial u_i^k}{\partial \beta^T}\right)|_{\beta=\hat{\beta}^k} \tag{4.9}$$

4.3.3 Additional Intuition

The multiple components behind the Robins and Wang imputation variance estimator can be intimidating. To gain intuition, it might be helpful to first consider a simple scenario where we estimate the variance for an analysis where all values are observed and thus no imputation. When no values are imputed, $\Delta = \Omega$ since:

$$S^{mis} = 0 \implies \kappa = 0 \implies \Delta = \Omega$$

Suppose we had a linear regression analysis model for the association between two variables, Y and X. Thus, $u_i = X_i(Y_i - X_i\beta)$ and $\tau = \frac{1}{n}\sum_{i=1}^{n} X_i^T X_i$. Our variance estimate is:

$$\begin{aligned}
\Gamma =& \frac{1}{n}\tau^{-1}\Omega(\tau^{-1})^T \\
=& \frac{1}{n}\left(\frac{1}{n}\sum X_i^T X\right)^{-1}\frac{1}{n}\sum X_i(Y - X_i\beta)X_i(Y - X_i\beta)\left(\frac{1}{n}\sum X_i^T X_i\right)^{-1} \\
=& \left(\sum X_i^T X_i\right)^{-1}\sum X_i(Y - X_i\beta)(Y - X_i\beta)X_i\left(\sum X_i^T X_i\right)^{-1} \\
=& \left(\sum X_i^T X_i\right)^{-1}\sum X_i Var(Y_i)X_i\left(\sum X_i^T X_i\right)^{-1}
\end{aligned}$$

We recognize this as the robust variance estimator for linear regression. In fact, the Robins and Wang variance estimator calculates the robust variance estimator when there is no imputed data.

When data are imputed, $\Delta \neq \Omega$ since $S^{mis} \neq 0$. As a result, our variance estimator calculation is no longer as simple as calculating the robust variance estimate based on the analysis model. This should make sense given the uncertainty from imputing values in addition to the uncertainty of generating estimates based on a sample. To better understand what is going on, it may be helpful to slightly re-arrange the Robins and Wang variance estimator as follows:

$$\begin{aligned}
\Gamma =& \frac{1}{n}\tau^{-1}\Delta(\tau^{-1})^T \\
=& \frac{1}{n}\tau^{-1}\Omega(\tau^{-1})^T \quad + \\
& \frac{1}{n}\tau^{-1}\kappa\Lambda\kappa^T(\tau^{-1})^T \quad + \\
& \frac{1}{n}\tau^{-1}\left(\frac{1}{n}\{\kappa d_i^T \bar{u_i} + (\kappa d_i^T \bar{u_i})^T\}\right)(\tau^{-1})^T
\end{aligned}$$

It should be emphasized that the second and third terms are calculated using components from both the imputation and analysis models. This contrasts with the calculation for the variance estimator proposed by Rubin that is based solely on the analysis model. This is advantageous when there is incompatibility between the imputation and analysis models.

4.4 Example 1: HST simulation

In their manuscript, HST calculated the Robins and Wang imputation variance estimator for four different scenarios of misspecification and incompatibility between imputation and analysis models. The R code provided below generates the simulation dataset described in their manuscript and reproduces the RW calculations for their first scenario. Please note that we have modified some notation for clarity.

4.4.1 Example 1: R code for data generation

Briefly, a hypothetical dataset contains 1000 subjects and five variables – natural log of insulin index (Y), weight (X), sex (Z_1=0 for males, Z_1=1 for females) age

(Z_2), and height (Z_3). One variable, weight, has observations missing completely random for 60% of males; weight observations are available for all women. Data were generated based on the following model:

$$Z_1 \sim \text{Bernoulli } (\pi)$$
$$Z_2, Z_3 \sim N(\alpha_0 + \alpha_1 * Z_1, \Sigma)$$
$$X \sim \zeta_0 + \zeta_1 Z_1 + \zeta_2 Z_2 + \zeta_3 Z_3 + \lambda error_w$$
$$Y \sim \gamma_0 + \gamma_1 X + \gamma_2 Z_1 + \gamma_3 Z_2 + \omega error_l$$

where $\pi = 0.4577$, $\alpha_0 = (25.02,\ 1.774)$, $\alpha_1 = (0,\ 0)$, $\Sigma = \begin{bmatrix} 0.5521 & 0.001574 \\ 0.001574 & 0.003705 \end{bmatrix}$,$\zeta_0 =$ -32.98, $\zeta_1 = 0$, ζ_2 = -0.01566, $\zeta_3 = 65.38$, λ=12.29, $\gamma_0 = 1.854$, γ_1=0.01119, $\gamma_2 = 0$, γ_3 =0.08003, ω=0.7887, and $error_w, error_l$ are normal error terms.

```
#Set seed for reproducibility
set.seed(455)
#Load relevant packages
        require(mvtnorm)

#Fixed values
        obs<-1000
        pi<-0.4577
        alpha0<-c(25.02,1.774)
        alpha1<-c(0,0)
        Sigma<-matrix(c(0.5521,0.001574,0.001574,0.003705),2,2)
        zeta<-c(-32.98,0,-0.01566,65.38)
        gamma<-c(1.854,0.01119,0,0.08003)
        lambda<-12.29
        omega<-0.7887
        error_w<-rnorm(obs)
        error_l<-rnorm(obs)
        prob.missing<-0.6
        Nimpute<-50     #Number of imputations

#Generate values
        z1<-rbinom(obs,1,pi)
        z2<-z3<-NULL
        for (i in 1:obs){
                z23<-rmvnorm(1, mean=alpha0 + z1[i]*alpha1,sigma=Sigma)
                z2[i]<-z23[,1]; z3[i]<-z23[,2]
        }

        X<-as.matrix(data.frame(1,z1,z2,z3))%*%t(t(zeta)) + lambda*error_w
```

```
        Y<-as.matrix(data.frame(1,X,z1,z2))%*%t(t(gamma)) + omega*error_1

#Randomly set X values to missing
        X.miss<-ifelse(z1==1,0, rbinom(obs,1,prob.missing))

#Original dataset
        ID<-1:obs
        dat<-data.frame(ID,Y,X,z1,z2,z3)
        dat$Intercept<-1
        dat$X<-ifelse(X.miss==1,NA,dat$X)
```

4.4.2 Example 1: R code for imputation model

Suppose that an imputer was instructed to implement a multiple imputation procedure to impute the missing weight measurements. For this procedure, the imputer fit a linear regression model with Y, Z_1, Z_2, and Z_3 as covariates using the observations with complete data. For each imputation, the imputer accounted for both parameter uncertainty and random noise in the predictions.

```
#Fit linear regression model using complete observations
        mod.impute<-glm(X~Y+z1+z2+z3,data=dat,family="gaussian")
        impute.vars<-c("Intercept","Y","z1","z2","z3")
        impvar_n<-length(impute.vars)

#Function to impute values
ImputationFn<-function(mod,datty){
        vars<-c("Intercept",all.vars(formula(mod)[-2]))
        newdatmat<-datty[,vars]

        #Account for parameter uncertainty
        #Re-draw from a multivariate normal distribution with mean and variance
        #as the parameter estimates and cov matrix from the regression model.
        desmatrix<-rmvnorm(1, mean=mod$coefficients, sigma=vcov(mod))
        linpred<-as.vector(desmatrix%*%t(newdatmat))

        #Account for random noise
        #Randomly sample from the residuals
        resid.y<-mod$residuals
        imputed.values<-linpred +
                as.numeric(sample(resid.y,length(linpred),replace=TRUE))
```

```
        return(imputed.values)
}


#Replace missing values with imputed value
        dat.imputed<-dat
        dat.imputed$Imputed<-with(dat.imputed,ifelse(is.na(X),1,0))

        imputed.values<-ImputationFn(mod.impute,dat.imputed)
        dat.imputed$X<-with(dat.imputed,ifelse(is.na(X),imputed.values,X))
```

4.4.3 Example 1: R code for analysis model

Now, suppose that the analyst is interested in estimating the association between log of the insulin index (Y) and weight (X) among males only ($Z_1 = 0$) by fitting a linear regression model that adjusts for Z_2. It should be noted that there is incompatibility between the imputation model and the analysis model since the imputations were generated using observations from both males and females while the analysis model is based on just males. The following code demonstrates how to fit this analysis model as well as extract each of the components from the analysis model that are necessary to compute the Robins and Wang imputation variance estimator.

```
dat.analysis<-dat.imputed[z1==0,] #males only
mod.analysis<-glm(Y~X + z2,data=dat.analysis,family="gaussian")
analy.vars<-names(mod.analysis$coef)
analy.vars<-ifelse(analy.vars=="(Intercept)","Intercept",analy.vars)
```

4.4.4 Example 1: R code for RW component calculations based on imputation model

The imputer needs to calculate and supply two datasets based on the score function of the imputation model, S_{mis} and d. The following code contains a function that will calculate and output S_{mis} and d for a linear regression imputation model for a single continuous variable. These calculations are performed for each imputation.

```
RW_Components_Imputation_Fn<-function(mod.impute){

        #Need to estimate sigma and calculate predicted values from output
```

```
sigma.est<-var(mod.impute$residuals)
pred.values<-predict(mod.impute,newdata=dat.imputed)

#Evaluate score function with respect to all parameters in the model
#This includes the the variance!
S_u<-(dat.imputed$X-pred.values)*dat.imputed[,impute.vars]/sigma.est
S_sigma<-0.5*(-1/sigma.est+ (dat.imputed$X-pred.values)^2/sigma.est^2)

#Preliminary matrix that indicates missingess for each subject
ImputedMat<-matrix(dat.imputed$Imputed==1,nrow(S_u),ncol(S_u),byrow=FALSE)
NOTImputedMat<-1-ImputedMat

#####################################################################
#Calculation of S_mis -- component from imputation model
#Output a dataset that is the evaluated score function for imputed obs
#####################################################################
S_u_imp<-S_u*ImputedMat
S_sigma_imp<-ifelse(dat.imputed$Imputed==1,S_sigma,0)
S_mis_imp<-cbind(S_u_imp,S_sigma_imp)

#####################################################################
#Calculation of D -- component from imputation model
#####################################################################
S_u_orig<-S_u*NOTImputedMat
S_sigma_orig<-ifelse(dat.imputed$Imputed==1,0,S_sigma)
S_orig<-cbind(S_u_orig,S_sigma_orig)

S2_uu<-matrix(NA,nrow(dat.imputed),length(impute.vars))
S2_uu<-(-1/sigma.est*
        t(dat.imputed[dat.imputed$Imputed==0,c(impute.vars)])%*%
        t(t(dat.imputed[dat.imputed$Imputed==0,c(impute.vars)])))
S2_usigma<-(-1/sigma.est^2*
        t(dat.imputed[dat.imputed$Imputed==0,c(impute.vars)])%*%
        (dat.imputed$X-pred.values)[dat.imputed$Imputed==0])
S2_sigmasigma<-(.5/sigma.est^2-(dat.imputed$X-pred.values)^2/sigma.est^3)

S2_sigmasigma<-ifelse(dat.imputed$Imputed==1,0,S2_sigmasigma)
D.inv<-matrix(0,impvar_n+1,impvar_n+1)
D.inv[1:impvar_n,1:impvar_n]<-S2_uu/nrow(dat.imputed)
D.inv[1:impvar_n,impvar_n+1]<-apply(S2_usigma,1,mean)/nrow(dat.imputed)
D.inv[impvar_n+1,1:impvar_n]<-apply(S2_usigma,1,mean)/nrow(dat.imputed)
D.inv[impvar_n+1,impvar_n+1]<-mean(S2_sigmasigma)
```

```
        Dmat<-solve(D.inv)
        d_t<-((-1)*Dmat)%*%t(S_orig)
        d<-t(d_t)

        return(list(S_mis_imp,d))
}
```

4.4.5 Example 1: R code for RW component calculations based on analysis model

The analyst needs to calculate and supply two datasets, u and τ. The following code contains a function that will calculate and output u and τ for a linear regression analysis model. These calculations are performed for each imputation.

```
RW_Components_Analysis_Fn<-function(mod.analysis){
        analysis.predict<-predict(mod.analysis)

        ######################################################################
        #Calculation of u
        #Evaluate the estimating equation for each observation
        ######################################################################
        U_imp_pre<-dat.analysis[,analy.vars]*
                                matrix((dat.analysis$Y-analysis.predict),
                                dim(dat.analysis[, analy.vars]))
        aa<-cbind(dat.analysis$ID,U_imp_pre)
        colnames(aa)<-c("ID",colnames(U_imp_pre))
        bb<-merge(dat.imputed[,c("ID","Imputed")],aa,by="ID",all=T)
        U_imp<-bb[,colnames(U_imp_pre)]
        U_imp[is.na(U_imp)]<-0

        ######################################################################
        #Calculation of tau
        #Take derivative of the est eq. and evaluate for each observation
        ######################################################################
        tau_imp<-t(dat.analysis[,analy.vars])%*%t(t(dat.analysis[,analy.vars]))


        return(list(U_imp,tau_imp))
}
```

4.4.6 Example 1: R code for RW multiple imputation variance calculation

The above components are calculated using one imputed dataset. The following code provides an outline for calculating the variance estimator across multiple imputations. For this example, the dataset was imputed 50 times. We output the variance estimates for the RW variance estimator and Rubin's variance estimator for comparison.

```
#Set seed for reproducibility
set.seed(455)
#Initialize select variables
    U_imp_sum<-0
    kappa_sum<-0
    tau_sum<-0
    analysis.est<-analysis.se<-NULL


#Impute values, perform analyses, and
#Calculate the relevant components necessary for RW for each imputation
for (p in 1:Nimpute){
        #Replace missing values with imputed value
        dat.imputed<-dat
        dat.imputed$Imputed<-with(dat.imputed,ifelse(is.na(X),1,0))
        imputed.values<-ImputationFn(mod.impute,dat.imputed)
        dat.imputed$X<-with(dat.imputed,ifelse(is.na(X),imputed.values,X))

        #Create analysis dataset based on inclusion criteria
        #Perform linear regression
        dat.analysis<-dat.imputed[z1==0,] #males only
        mod.analysis<-glm(Y~X + z2,data=dat.analysis,family="gaussian")
        analy.vars<-names(mod.analysis$coef)
        analy.vars<-ifelse(analy.vars=="(Intercept)","Intercept",analy.vars)

        #Capture estimates and corresponding SE from each imputation
        analysis.est[p]<-mod.analysis$coef["X"]
        analysis.se[p]<-sqrt(diag(vcov(mod.analysis))["X"])

        #Evaluate all four components for a given imputation
        ImputationComponents<-RW_Components_Imputation_Fn(mod.impute)
        AnalysisComponents<-RW_Components_Analysis_Fn(mod.analysis)
        S_mis_imp<-ImputationComponents[[1]]
        d<-ImputationComponents[[2]] #Only need to calculate d once
```

```
        U_imp<-AnalysisComponents[[1]]
        U_imp[is.na(U_imp)]<-0
        tau_imp<-AnalysisComponents[[2]]

        #Create summations of certain components across imputations
        U_imp_sum<-U_imp_sum+U_imp
        tau_sum<-tau_sum+tau_imp


        #Calculate kappa for a given imputation
        #Note that S_mis_imp does not change across imputations
        kappa_imp<-t(U_imp)%*%t(t(S_mis_imp))
        #Create summation for kappa across imputations
        kappa_sum<-kappa_sum+kappa_imp
}
#################################################################### #
#Combine components together to calculate Robins and Wang variance estimator
#################################################################### #
        u_bar<-U_imp_sum/Nimpute
        u_bar<-t(t(u_bar))

        omega<-(t(u_bar)%*%t(t(u_bar)))/(obs)
        kappa<-t(t(kappa_sum/(obs*Nimpute)))

        alpha<-(t(d)%*%d)/obs

        delta<-omega + kappa%*%alpha%*%t(kappa) +
                (1/obs)*(kappa%*%t(d)%*%u_bar + t(kappa%*%t(d)%*%u_bar))
        tau<-tau_sum/(Nimpute*obs)
        GAMMA<-(1/obs)*t(t(solve(tau)))%*%delta%*%t(solve(tau))

        #Save SE estimates to calculate CIs!
        RobinsWangSE<-sqrt(diag(GAMMA)["X"])
        RubinSE<-sqrt( mean(analysis.se^2)+ (Nimpute+1)/Nimpute*var(analysis.est))


        #Output imputation estimate for parameter
        mean(analysis.est)

## [1] 0.01101867

        #Output variance estimates for RW and Rubin (X 1000)
        RobinsWangSE^2 *1000
```

```
##           X
## 0.008527562

        RubinSE^2 * 1000

## [1] 0.01429272
```

4.4.7 Example 1: Results

In this example, the imputation estimate for the weight coefficient was 0.0110 with corresponding variance using Robins and Wang's approach, 0.0085. This estimate was smaller than the estimated variance based on Rubin's approach, 0.0143. Please note that due to small variance sizes, the values reported were multipled by 1000.

To compare our findings with those provided in the HST manuscript, we repeated the simulation 2500 times. The mean estimates for the weight coefficient (0.0113 vs. 0.0112) and Rubin variance (0.0122 vs. 0.0123) were similar to those reported in the HST manuscript. We note that the empirical variance (0.0073 vs. 0.0007) and mean estimated RW variance (0.0073 vs. 0.0007) are higher than those reported in the original manuscript by a factor of 10. However, given that the coverage estimates for both the RW (0.948 vs. 0.948) and RR (0.986 vs. 0.990) approaches are nearly identical to those reported in the manuscript, we suspect that there was a typo in the HST paper and our results are accurate.

For this example, the imputation variance estimator proposed by Robins and Wang had better coverage and was closer to the empirical variance estimate than the estimated based on Rubin's approach. This is not surprising given the incompatibility between the imputation and analysis models.

4.5 Example 2: EHR example

In Chapter 2, we discussed source data verification (SDV) for a subset of electronic health records and the impact on corresponding data analyses. For that study, we maintained a pre-audit dataset containing the originally entered values for all records and an audited dataset containing updated values for a subset of records based on the SDV. Given a dataset with measurement error and a validation subsample, we can consider this a missing data problem that can be addressed using multiple imputation.

We now use this framework to demonstrate that the Robins and Wang variance estimator can be calculated in settings beyond the simplified scenario described in the first example. In the following example, we increase the percentage of missing values to 75%, we impute values for multiple variables (including continuous and binary variables), and we use a logistic regression analysis model. Additionally, we add a practical consideration that is common for many analyses using observational datasets - a non-static exclusion criteria. Inclusion and exclusion criteria are non-static when they are based on an imputed variable. In these settings, a subject may be excluded in some imputation iterations but not others.

4.5.1 Example 2: R code for data generation

For this example, suppose we have a dataset containing 4000 electronic health records with four key variables: two continuous, correlated variables, X_1 and X_2, drawn from a bivariate normal distribution with mean 0, variance 1, and covariance -0.25; a continuous variable $\mathcal{A}^*$ drawn from a normal distribution with mean 1 and variance 1, and a binary variable $\mathcal{D}^*$ drawn from a Bernoulli distribution with the logit probability of success equal to $-3 + 0.5\mathcal{A}^*$. Suppose $\mathcal{A}^*$, $\mathcal{D}^*$ are error-prone versions of the actual variables of interest, $\mathcal{A}$, $\mathcal{D}$. $\mathcal{A}$ was drawn from a Normal distribution with mean equal to $-X_1 + 0.5X_2 + 0.9\mathcal{A}^* + 0.5\mathcal{D}^*$ and variance 2. Finally, $\mathcal{D}$ is drawn from a Bernoulli distribution with the logit probability of success equal to $-5.5 - 2X_1 + X_2 + 5\mathcal{D}^* + 0.5\mathcal{A}$. Note that we assume that there are no data errors for the other two variables, X_1 and X_2. Note also that we sample $\mathcal{A}$ and $\mathcal{D}$ conditional on their error-prone counterparts, $\mathcal{A}^*$ and $\mathcal{D}^*$, for ease of properly specifying the model.

A subset of 1000 subjects were randomly selected to represent an audited cohort with ($\mathcal{A}$, $\mathcal{D}$) known; for the remaining 3000 subjects, $\mathcal{A}$, $\mathcal{D}$ were treated as missing. X_1 and X_2 were treated as known for all 4000 subjects.

```
set.seed(455)

#Load relevant packages
        require(mvtnorm)

#Fixed values
        obs<-4000
        n.sample<-1000
        ID<-rep(1:obs)
        XYcov<-(-0.25)
        Xvar<-1
        Yvar<-1
        Astar.var<-1
        A.var<-1
        beta.star<-c(1,0,0)
        gamma.star<-c(-3,0,0,0.5)
        beta<-c(0,-1,0.5,.9,0.5)
        gamma<-c(-5.5,-2,1,0.0,5.0,0.5)
        Nimpute<-50 #Number of imputations
        exclusionthreshold<-2

#Generate values
#Covariates X and Y
        XY<-rmvnorm(obs, mean=c(0,0),sigma=matrix(c(Xvar,XYcov,XYcov,Yvar),2,2))
        X<-XY[,1]
        Y<-XY[,2]
        Xmat<-cbind(1,X,Y)
        rownames(Xmat)<-ID
        colnames(Xmat)<-c("Intercept","X","Y")
####################################
#Unvalidated data
####################################
#A*
        A.star<-rnorm(obs,Xmat%*%beta.star,Astar.var)
#D*
        LP.D.star<-Xmat%*%gamma.star[1:ncol(Xmat)] +
                                A.star*gamma.star[length(gamma.star)]
        p.D.star<-exp(LP.D.star)/(1+ exp(LP.D.star))
        #Indicator of event (D*) at time t
        D.star<-rbinom(obs,1,p.D.star)
####################################
#Validated data
####################################
```

```
#A
        LP.A<-Xmat%*%beta[1:ncol(Xmat)] + A.star*beta[ncol(Xmat)+1] +
                        D.star*beta[ncol(Xmat)+2]
        A<-rnorm(obs,LP.A,A.var)
#D
        LP.D<-Xmat%*%gamma[1:ncol(Xmat)]  + A.star*gamma[ncol(Xmat)+1]  +
                        D.star*gamma[ncol(Xmat)+2] +A*gamma[ncol(Xmat)+3]
        p.D<- exp(LP.D)/(1+ exp(LP.D))
        #Indicator of event (D) at time t
        D<-rbinom(obs,1,p.D)

#Randomly sample records to represent audited cohort
        ChooseSubset<-sample(1:obs,n.sample,replace=F)
        sampled<-as.numeric((1:obs)%in%ChooseSubset)

#Original dataset
        dat<-dat.original<-data.frame(ID,X,Y,A.star,D.star,A,D)
        dat$Intercept<-1
        dat$A<-ifelse(sampled==0,NA,dat$A)
        dat$D<-ifelse(sampled==0,NA,dat$D)
```

4.5.2 Example 2: R code for imputation

In this setting, we specify a joint imputation model for $(\mathcal{A}, \mathcal{D})$ by fitting a linear regression model for $\mathcal{A}$ conditional on $\mathcal{A}^*$,$\mathcal{D}^*$, X_1 and X_2 and a logistic regression model for $\mathcal{D}$ conditional on $\mathcal{A}^*$, $\mathcal{D}^*$, X_1, X_2, and $\mathcal{A}$ using the subset of 1000 audited records. For each imputation, the imputer accounted for both parameter uncertainty and random noise in the predictions.

```
dat.imputed<-dat
dat.imputed$Imputed<-as.numeric(sampled==0)
#Fit linear regression model using complete observations
    modA<-glm(A~X+Y+A.star+D.star,data=dat,family="gaussian",y=FALSE,model=FALSE)
    modD<-glm(D~X+Y+A.star+D.star+A,data=dat,family="binomial",y=FALSE,model=FALSE)


#Function to impute values
ImputationFn<-function(mod,datty){
vars<-c("Intercept",all.vars(formula(mod)[-2]))
newdatmat<-datty[,vars]
```

```
        #Account for parameter uncertainty
        #Re-draw from a multivariate normal distribution with mean and variance
         #as the parameter estimates and cov matrix from the regression model.
        desmatrix<-rmvnorm(1, mean=mod$coefficients, sigma=vcov(mod))
        linpred<-as.vector(desmatrix%*%t(newdatmat))

        if (mod$family[1]=="binomial"){ return(linpred)}
        if (mod$family[1]=="gaussian"){
        #Account for random noise
        #Randomly sample from the residuals
                resid.y<-mod$residuals
                imputed.values<-linpred
                        + as.numeric(sample(resid.y,length(linpred),replace=TRUE))
        }
        return(imputed.values)
}

        #Replace missing balues with imputed value
        imputed.A<-ImputationFn(modA,datty=dat.imputed)
        dat.imputed$A[sampled==0]<-imputed.A[sampled==0]
        imputed.LP.D<-ImputationFn(modD,datty=dat.imputed)
        imputed.D<-rbinom(length(imputed.LP.D),1,
                                exp(imputed.LP.D)/(1+exp(imputed.LP.D)))
        dat.imputed$D[sampled==0]<-imputed.D[sampled==0]
```

4.5.3 Example 2: R code for analysis model

To estimate the association between a predictor variable $\mathcal{A}$ and a binary outcome $\mathcal{D}$, we fit a logistic regression analysis model using the imputed dataset. Note that for this analysis we exclude subjects with $\mathcal{A} \leq 2$. Thus, some records incorporated in the imputation model are not included in the analysis model. It should be noted that there is incompatibility between the imputation model and the analysis model since the imputations were generated using observations from all subjects while the analysis model was based on just those with an imputed value of $\mathcal{A} > 2$. The following code demonstrates how to fit this analysis model.

```
#Generate dataset where obs meet inclusion criteria
dat.analysis<-dat.imputed[dat.imputed$A>exclusionthreshold,]
mod.analysis<-glm(D~A,data=dat.analysis,family="binomial")
```

```
analy.vars<-names(mod.analysis$coef)
analy.vars<-ifelse(analy.vars=="(Intercept)","Intercept",analy.vars)
```

4.5.4 Example 2: R code for RW component calculations based on imputation model

The imputer needs to calculate and supply two datasets based on the score function of the imputation model, S_{mis} and d. The following code contains several functions. The first two functions evaluate the score function or its derivative for each subject. We then expand the function that was provided in the first example to calculate and output the two components (S_{mis} and d) for the RW variance estimator from the imputation model. Here, we allow for a joint imputation model and the variable being imputed can be binary or continuous. This function is intended to be used for each imputation.

```
#Function to evaluate score function
#Code here works when imputation model is linear or logistic regression
S_u_function<-function(datty,mod.impute,imputedvar){
        imputemod.vars<-c("Intercept",all.vars(formula(mod.impute)[-2]))

if (mod.impute$family[1]=="gaussian"){
        sigma.est<-var(mod.impute$residuals)
        pred.values<-predict(mod.impute,newdata=datty)
        S_u_1<-(imputedvar-pred.values)*datty[,imputemod.vars]/sigma.est
        S_sigma_1<-0.5*(-1/sigma.est+ (imputedvar-pred.values)^2/sigma.est^2)
        S_u<-cbind(S_u_1,S_sigma_1)
        modelvarcount<-length(imputemod.vars)+1 #Account for sigma
}

if (mod.impute$family[1]=="binomial"){
        #pred.values is actually predicted LP
        pred.values<-predict(mod.impute,newdata=datty,returnvar="linpred")
        S_u<-datty[,imputemod.vars]*
                        (imputedvar-exp(pred.values)/(1+exp(pred.values)))
        modelvarcount<-length(imputemod.vars)
}
return(S_u)
}

#Function to evaluate derivative of score function
```

```
#Code here works when imputation model is linear or logistic regression
S2_function<-function(datty,mod.impute){
    imputemod.vars<-c("Intercept",all.vars(formula(mod.impute)[-2]))
    impvar_n<-length(imputemod.vars)

if (mod.impute$family[1]=="gaussian"){
        sigma.est<-var(mod.impute$residuals)
        pred.values<-predict(mod.impute,newdata=datty)

        S2_uu<-matrix(NA,nrow(datty),length(imputemod.vars))
        S2_uu<-with(datty,(-1/sigma.est*t(datty[Imputed==0,c(imputemod.vars)])%*%
                t(t(datty[Imputed==0,c(imputemod.vars)]))))
        S2_usigma<-with(datty,(-1/sigma.est^2*t(datty[Imputed==0
                ,c(imputemod.vars)])%*%(X-pred.values)[Imputed==0]))
        S2_sigmasigma<-sum(with(datty,ifelse(Imputed==1,0,
                1/(2*sigma.est^2)-(X-pred.values)^2/sigma.est^3)))
        S2_mod<-matrix(0,impvar_n+1,impvar_n+1)
        S2_mod[1:impvar_n,1:impvar_n]<-S2_uu/nrow(datty)
        S2_mod[1:impvar_n,impvar_n+1]<-apply(S2_usigma,1,mean)/nrow(datty)
        S2_mod[impvar_n+1,1:impvar_n]<-apply(S2_usigma,1,mean)/nrow(datty)
        S2_mod[impvar_n+1,impvar_n+1]<-S2_sigmasigma/nrow(datty)
}
if (mod.impute$family[1]=="binomial"){
        ImputeX<-datty[,imputemod.vars]
        #pred.values is actually predicted LP
        pred.values<-predict(mod.impute,newdata=datty,type="link")
        S2_mod<-matrix(NA,ncol(ImputeX),ncol(ImputeX))
        constant.term<-exp(pred.values)/(1+exp(pred.values))^2
        for (matrow in 1:ncol(ImputeX)){
                for (matcol in 1:ncol(ImputeX)){
                temp<-(-1)*(ImputeX[,matrow])*((ImputeX[,matcol]))*constant.term
                temp[datty$Imputed==1]<-0
                S2_mod[matrow,matcol]<-sum(temp)/length(datty$ID)
          }
        }
}
  return(S2_mod)
}

#Function that calculates and outputs the required components from imputation model
#Updated from Example 1 to allow for joint imputation models
RW_Components_Imputation_Fn<-function(){
```

```
        #Evaluate score function with respect to all parameters in the models
        S_u_mod1<-S_u_function(dat.imputed,modA,dat.imputed$A)
        S_u_mod2<-S_u_function(dat.imputed,modD,dat.imputed$D)
        S_u<-data.frame(S_u_mod1,S_u_mod2)

        #Preliminary matrix that indicates missingess for each subject
        ImputedMat<-matrix(dat.imputed$Imputed==1,nrow(S_u),ncol(S_u),byrow=FALSE)
        NOTImputedMat<-1-ImputedMat

        ######################################################################
        #Calculation of S_mis -- component from imputation model
        #Output a dataset that is the evaluated score function for imputed obs
        ######################################################################
        S_mis_imp<-S_u*ImputedMat

        ######################################################################
        #Calculation of D -- component from imputation model
        ######################################################################
        S_orig<-S_u*NOTImputedMat
        S2_mod1<-S2_function(dat.imputed,modA)
        S2_mod2<-S2_function(dat.imputed,modD)
        n_S2mod1<-ncol(S2_mod1)
        n_S2mod2<-ncol(S2_mod2)
        n_S2mod<-n_S2mod1+n_S2mod2

        S2<-matrix(0,n_S2mod,n_S2mod)
        S2[1:n_S2mod1,1:n_S2mod1]<-S2_mod1
        S2[(n_S2mod1+1):n_S2mod,(n_S2mod1+1):n_S2mod]<-S2_mod2

        Dmat<-solve(S2)
        d_t<-((-1)*Dmat)%*%t(S_orig)
        d<-t(d_t)

        return(list(S_mis_imp,d))
}
```

4.5.5 Example 2: R code for RW component calculations based on analysis model

The analyst needs to calculate and supply two datasets, u and τ. The following code contains a function that will calculate and output u and τ for a logistic regression analysis model. Note that our analysis model for this example (logistic regression) is different than the analysis model in the first example (linear regression). As a result,

the code to calculate u and τ has been changed. These calculations are performed for each imputation.

```
RW_Components_Analysis_Fn<-function(){
        analysis.predict<-predict(mod.analysis,type="response")

        ###################################################################
        #Calculation of u
        #Evaluate the estimating equation for each observation
        ###################################################################
        U_imp_pre<-dat.analysis[,analy.vars]*
                                matrix((dat.analysis$D-analysis.predict),
                                dim(dat.analysis[, analy.vars]))
        aa<-cbind(dat.analysis$ID,U_imp_pre)
        colnames(aa)<-c("ID",colnames(U_imp_pre))
        bb<-merge(dat.imputed[,c("ID","Imputed")],aa,by="ID",all=T)
        U_imp<-bb[,colnames(U_imp_pre)]
        U_imp[is.na(U_imp)]<-0

        ###################################################################
        #Calculation of tau
        #Take the derivative of the estimating equation and evaluate for each obs
        ###################################################################
        tempdat<-dat.analysis[,analy.vars]
        #pred.values is actually predicted LP
        analysis.predLP<-predict(mod.analysis,newdata=tempdat,type="link")
        tau_imp<-matrix(NA,ncol(tempdat),ncol(tempdat))
        rownames(tau_imp)<-colnames(tau_imp)<-colnames(tempdat)
        constant.term<-exp(analysis.predLP)/(1+exp(analysis.predLP))^2
        for (matrow in 1:ncol(tempdat)){
                for (matcol in 1:ncol(tempdat)){
                temp<-(-1)*(tempdat[,matrow])*((tempdat[,matcol]))*constant.term
                tau_imp[matrow,matcol]<-(-1)*sum(temp)
          }
        }
        return(list(U_imp,tau_imp))
}
```

4.5.6 Example 2: R code for RW multiple imputation variance calculation

The above components are calculated using one imputed dataset. The following code provides an outline for calculating the variance estimator across multiple impu-

tations. For this example, the dataset was imputed 50 times. We output the variance estimates for the RW variance estimator and RR variance estimator for comparison.

```
#Initialize select variables
    U_imp_sum<-0
    kappa_sum<-0
    tau_sum<-0
    analysis.est<-analysis.se<-analysisN<-NULL
for (p in 1:Nimpute){
        #Replace missing values with imputed value
        dat.imputed<-dat
        dat.imputed$Imputed<-as.numeric(sampled==0)

        imputed.A<-ImputationFn(modA,datty=dat.imputed)
        dat.imputed$A[sampled==0]<-imputed.A[sampled==0]
        imputed.LP.D<-ImputationFn(modD,datty=dat.imputed)
        imputed.D<-rbinom(length(imputed.LP.D),1,exp(imputed.LP.D)/
                (1+exp(imputed.LP.D)))
        dat.imputed$D[sampled==0]<-imputed.D[sampled==0]

        #Create analysis dataset based on inclusion criteria
        #Perform logistic regression
        dat.analysis<-dat.imputed[dat.imputed$A>exclusionthreshold,]
        mod.analysis<-glm(D~A,data=dat.analysis,family="binomial")
        analy.vars<-names(mod.analysis$coef)
        analy.vars<-ifelse(analy.vars=="(Intercept)","Intercept",analy.vars)

        #Capture estimates and corresponding SE from each imputation
        analysis.est[p]<-mod.analysis$coef["A"]
        analysis.se[p]<-sqrt(diag(vcov(mod.analysis))["A"])

        #Evaluate all four components for a given imputation
        ImputationComponents<-RW_Components_Imputation_Fn()
        AnalysisComponents<-RW_Components_Analysis_Fn()
        S_mis_imp<-ImputationComponents[[1]]
        d<-ImputationComponents[[2]] #Only need to calculate d once
        U_imp<-AnalysisComponents[[1]]
        U_imp[is.na(U_imp)]<-0
        tau_imp<-AnalysisComponents[[2]]

        #Create summations of certain components across imputations
        U_imp_sum<-U_imp_sum+U_imp
        tau_sum<-tau_sum+tau_imp
```

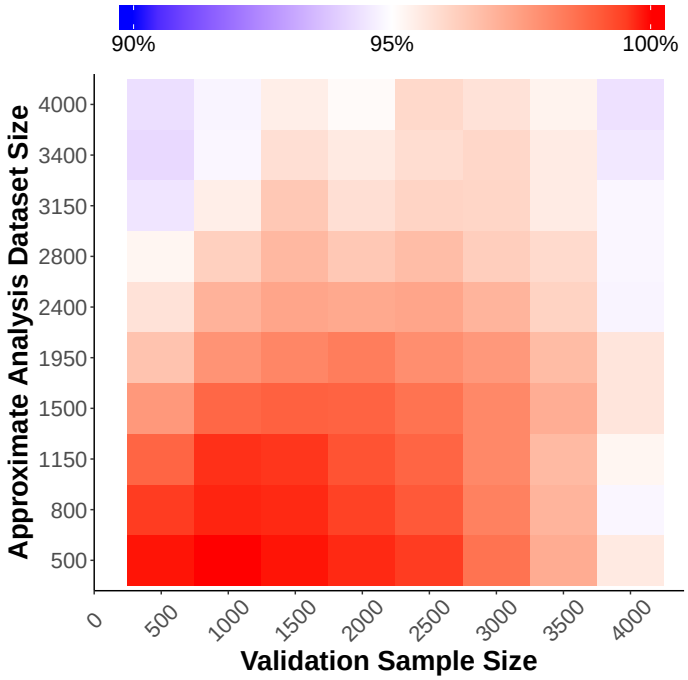
```
        #Calculate kappa for a given imputation
        kappa_imp<-t(U_imp)%*%t(t(S_mis_imp))
        #Create summation for kappa across imputations
        kappa_sum<-kappa_sum+kappa_imp

        #Save the number of observations in the analysis dataset
        analysisN[p]<-nrow(dat.analysis)
}
################################################################ #
#Combine components together to calculate Robins and Wang variance estimator
################################################################ #
        u_bar<-U_imp_sum/Nimpute
        u_bar<-t(t(u_bar))

        omega<-(t(u_bar)%*%t(t(u_bar)))/(obs)
        kappa<-t(t(kappa_sum/(obs*Nimpute)))

        alpha<-(t(d)%*%d)/obs

        delta<-omega + kappa%*%alpha%*%t(kappa) +
                (1/obs)*(kappa%*%t(d)%*%u_bar + t(kappa%*%t(d)%*%u_bar))
        tau<-tau_sum/(Nimpute*obs)
        GAMMA<-(1/obs)*t(t(solve(tau)))%*%delta%*%t(solve(tau))

        #Save SE estimates to calculate CIs!
        RobinsWangSE<-sqrt(diag(GAMMA)["A"])
        RubinSE<-sqrt( mean(analysis.se^2)+ (Nimpute+1)/Nimpute*var(analysis.est))

        #Output imputation estimate for parameter
        mean(analysis.est)

## [1] 1.081653

        #Output variance estimates for RW and Rubin
        RobinsWangSE^2

##         A
## 0.0101567

        RubinSE^2

## [1] 0.01374821
```

4.5.7 Example 2: Results

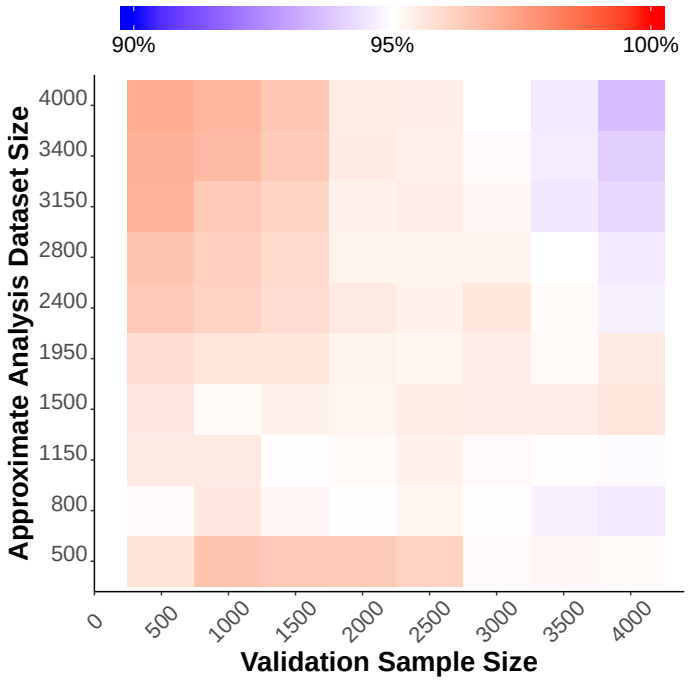
The imputation estimate for the coefficient was 1.0817 with corresponding variance using Robins and Wang's approach, 0.0102. For comparison, the estimated variance based on Rubin's approach was 0.0137. Across 50 imputations, the average analysis dataset size was 1044 after excluding subjects with $\mathcal{A} \leq 2$. The smallest and largest analysis datasets were 989 and 1091 subjects, respectively.

When we repeated the simulation for this example 2500 times, the mean RW variance estimate (0.0861) was smaller than the mean RR variance estimate (0.1152). For comparison, the empirical variance estimate was (0.0068). The imputation variance estimator proposed by Robins and Wang had better coverage (0.9556 vs. 0.9968) and was closer to the empirical variance estimate than the estimate based on Rubin's approach. These findings suggest a large discrepancy between the RW and RR variance estimates when the percentage of missing observations is high (75%) and a small proportion of observations are included in the analysis model (26.1%).

To better compare the performance of the two variance estimators, we expanded the simulation to include different validation subsample sizes (n= 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000) as well as different inclusion thresholds ($\mathcal{A} > \{-\infty$,-1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3$\}$). For each inclusion threshold, we calculated the average size of the corresponding analysis dataset that remained after subjects were excluded. The average analysis dataset size varied from 524 to 4000. For each simulation, we calculated 95% Wald confidence intervals using the RW imputation variance and the RR imputation variance estimates. Coverage estimates based on 2500 simulations using both RR (Table 4.1) and RW (Table 4.2) for all 80 original combinations of validation subsample sizes and inclusion thresholds are provided in Appendix C. To illustrate how both imputation variance estimators perform as the validation subsample and amount of incompatibility changes, we generated heat maps (Figure 4.1) corresponding to the confidence interval coverage using RW and RR for each combination. Coverage estimates were interpolated by smoothing across all possible combinations of validation sample size and analysis dataset size. The coverage probability gets higher for confidence intervals calculated using Rubin's imputation variance estimator as more subjects are excluded from the analysis dataset and more observations are imputed.



(a) Rubin



(b) Robins and Wang

Figure 4.1: Coverage estimates for 95% Wald confidence intervals calculated using Rubin or Robins & Wang variance estimates for combinations of validation subsample sizes and analysis dataset sizes. Plots were generated for combinations of 8 validation subsample sizes (n= 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000) and 10 different inclusion thresholds ($\mathcal{A} > \{-\infty$,-1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3$\}$. For each inclusion threshold, we calculated the average of the corresponding analysis dataset that was generated. Coverage estimates were interpolated using smoothing for all other values.

4.6 Example 3: TDMI example

In Chapter 3, we proposed a time-discretized modeling and imputation (TDMI) approach to simultaneously account for errors in both predictor and time-to-event outcome variables. In the example from that chapter, the available data for all subjects is error-prone and validated data corresponding to the same variables is available for a subset of subjects. This again is a setting with missing data that can be addressed using multiple imputation. For this TDMI approach, imputation models were based on person-month observations that were generated by discretizing person-level data into monthly intervals for key variables. After imputation, observations were "undiscretized" and analysis models were based on person-level observations. For that study, we were interested in time-to-event analyses using both Cox regression and Kaplan-Meier estimators.

In the original analysis for this study, we calculated confidence intervals based on RW's variance estimator. This decision was motivated by the incompatibility between the imputation and analysis models as a result of differences in the unit of observation (person vs. person-month) and subjects excluded in the analysis model. As part of this analysis, we also conducted a simulation study to assess the impact of misspecification in the imputation model. Considering this simulation involved the implementation of the RW variance estimator in a complex study design - time-discretized data, joint imputation models, non-static exclusion criteria, and time-to-event outcomes - we provide here the statistical code to reproduce a portion of this simulation.

4.6.1 Example 3: R code for data generation

Simulated data were based on a simplified version of the dataset described in Chapter 3 where we were interested in the association between a predictor and a time-to-event outcome and the incidence of that outcome at a fixed timepoint. The simulated cohort included 4000 subjects each with 100 months of follow-up. Each subject was assigned two continuous, correlated variables, X_1 and X_2 drawn from a bivariate normal distribution with mean 0, variance 1, and covariance -0.25. For simplicity, these variables were time-invariant. $\mathcal{A}^*_m$ was drawn from a Bernoulli distribution at month $m = 1, \ldots, 100$ with the logit probability of success equal to $-3 - 0.02m$. $\mathcal{D}^*_m$ was drawn from a Bernoulli distribution with the logit probability of success equal to

$-5-0.02m+0.5\mathcal{A}^*_m$. $\mathcal{A}_m$ was then drawn from a Bernoulli distribution with the logit probability of success equal to $-5-0.02m-X_1+0.5X_2+4\mathcal{A}^*_m+0.5\mathcal{D}^*_m$. Finally, $\mathcal{D}_m$ was drawn from a Bernoulli distribution with the logit probability of success equal to $-7-0.02m-2X_1+X_2+4\mathcal{D}^*_m+0.5\mathcal{A}_m$.

For all 4000 subjects, we derive several variables for analysis. For example, in the original study, we were interested in the time from treatment initiation to the first event. We are interested in the number of months between the first instance of $\mathcal{A}_m = 1$ and $\mathcal{D}_m = 1$. T_0 and T_E are computed as the smallest values of m with $\mathcal{A}_m = 1$ and $\mathcal{D}_m = 1$, respectively. If $\mathcal{A}_m = 0$ (or similarly $\mathcal{D}_m = 0$) for all m, then T_0 (similarly T_E) was set to an arbitrary value bigger than 100 (e.g., 101). Then $Y = min(T_E, 100) - T_0$ and $D = I(T_E \leq 100)$. Records were eligible for analysis if $W = I(T_0 \leq 100)I(T_0 < T_E) = 1$. W^*, Y^*, and D^* were similarly computed using $\overline{\mathcal{A}^*}$ and $\overline{\mathcal{D}^*}$. Since X_1 and X_2 are time invariant, they stay the same for each subject. The parameters of interest were, among those with $W = 1$, $P(T_E - T_0 \leq 60)$ and β from the proportional hazards model, $\lambda(m|X_1) = \lambda_0(m)exp(\beta X_1)$.

A subset of 1000 subjects were randomly selected to represent an audited cohort with $\overline{\mathcal{A}}$ and $\overline{\mathcal{D}}$ (and hence, W, Y, and D) known; for the remaining 3000 subjects, $\overline{\mathcal{A}}$ and $\overline{\mathcal{D}}$ (and therefore W, Y, and D) were treated as missing. $\overline{\mathcal{A}^*}, \overline{\mathcal{D}^*}, X_1$, and X_2 were treated as known for all 4000 subjects. The TDMI procedure was implemented to multiply impute missing values of $\overline{\mathcal{A}}$ and $\overline{\mathcal{D}}$ and then to derive (W, Y, D) for the 3000 subjects. The imputation models were correctly specified models for $\overline{\mathcal{A}}$ and $\overline{\mathcal{D}}$ that included X_1 and X_2. The parameters of interest were estimated using Kaplan-Meier estimates and Cox regression applied to the multiply imputed data.

```
set.seed(455)

#Load relevant packages
        require(plyr)
        require(survival)
        require(mvtnorm)
        require(MASS)
        require(rms)
        require(data.table)

#Fixed values
        n<-4000
        n.sample<-1000
        beta.unval<-c(-3,-0.02,0,0)
```

```
        gamma.unval<-c(-5,-0.02,0,0,0.5)
        beta<-c(-5,-0.02,-1,0.5,4,0.5)
        gamma<-c(-7,-0.02,-2,1,0.0,4.0,0.5)
        XYcov<-(-0.25)
        Xvar<-1
        Yvar<-1
        maxtime<-time<-100 #total number of months
        times<-round((0:maxtime)*30.437/365.25,3)
        times<-times[c(-1,-length(times))] #vector of timepoints in month intervals
        evaltime<-5 #timepoint of interest in KM analysis
        CoxVar<-"X" #variable name of key parameter in cox analysis
        Nimpute<-20

#Generate values
        ID<-rep(1:n,time)
        ID<-paste("A",formatC(rep(1:n,time),
                width=6,format='f',digits=0,flag='0'),sep="")
        Intercept<-rep(1,n)

#Covariates X and Y
        XY<-rmvnorm(n, mean=c(0,0),sigma=matrix(c(Xvar,XYcov,XYcov,Yvar),2,2))
        X<-XY[,1]
        Y<-XY[,2]
        Xmat<-cbind(1,sort(rep(1:maxtime,n)),X,Y)
        colnames(Xmat)<-c("Intercept","time","X","Y")

####################################
#Unvalidated data
####################################
#A*
        LP.A.unval<-Xmat%*%beta.unval
        p.A.unval<- exp(LP.A.unval)/(1+ exp(LP.A.unval))
        #Indicator of A at time t
        e.A.unval<-rbinom(n*maxtime,1,p.A.unval)
        A.star<-matrix(e.A.unval,n,maxtime,byrow=FALSE)
#D*
        LP.D.unval<-Xmat%*%gamma.unval[1:4] + e.A.unval*gamma.unval[5]
        p.D.unval<-exp(LP.D.unval)/(1+ exp(LP.D.unval))
        #Indicator of event (D) at time t
        e.D.unval<-rbinom(n*time,1,p.D.unval)
        D.star<-matrix(e.D.unval,n,maxtime,byrow=FALSE)

####################################
```

```
######Validated data
####################################
#A
        LP.A<-Xmat%*%beta[1:4] + e.A.unval*beta[5] + e.D.unval*beta[6]
        p.A<- exp(LP.A)/(1+ exp(LP.A))
        #Indicator of A at time t
        e.A<-rbinom(n*maxtime,1,p.A)
        A<-matrix(e.A,n,maxtime,byrow=FALSE)
#D
        LP.D<-Xmat%*%gamma[1:4] + e.A.unval*gamma[5] +
                        e.D.unval*gamma[6]+e.A*gamma[7]
        p.D<- exp(LP.D)/(1+ exp(LP.D))
        #Indicator of event (D) at time t
        e.D<-rbinom(n*maxtime,1,p.D)
        D<-matrix(e.D,n,maxtime,byrow=FALSE)

#Randomly sample records to represent audited cohort
        ChooseSubset<-sample(1:n,n.sample,replace=F)
        sampled<-as.numeric((1:n)%in%ChooseSubset)


#Combine generated values to create a person-month dataset of SIMULATED values
        dat.simulated<-data.frame(ID,sampled,cbind(sort(rep(1:maxtime,n)),
                Intercept,X,Y,as.vector(A.star),as.vector(D.star)),
                as.vector(A),as.vector(D))
        colnames(dat.simulated)<-c("ID","sampled","time","Intercept","X","Y",
                "A.star","D.star","A","D")

#Original dataset
        dat.simulated<-dat.simulated[order(dat.simulated$ID,dat.simulated$time),]
        alldata<-dat.simulated
        #Set values to missing for A and D for the records that were not audited
        alldata$A[alldata$sampled==0]<-NA
        alldata$D[alldata$sampled==0]<-NA

        alldata$AGE_AT_LAST_VISIT<-maxtime*30.437/365.25

#The data as simulated + imputed is person-month data
#For analysis, we want to analysize person-level data of time-to-event outcomes
#The following function undiscretizes data and creates one observation per subject
BuildAnalysisDat<-function(dat){

        #Create variables corresponding to imputed A,D
```

```
#Assign new value if imputed; Assign original value if not imputed
dat$A.imputed<-dat$A
dat$D.imputed<-dat$D
dat$A.imputed[dat$sampled==0]<-as.vector(dat$pred.A)[dat$sampled==0]
dat$D.imputed[dat$sampled==0]<-as.vector(dat$pred.D)[dat$sampled==0]

#For each person, determine first month that imputed A=1
FirstAdat<-data.frame(setDT(dat[dat$A.imputed==1 &
        is.na(dat$A.imputed)==F,c("ID","time")])[, lapply(.SD, min),
        by = dat[dat$A.imputed==1 & is.na(dat$A.imputed)==F,
        c("ID","time")]$ID])
colnames(FirstAdat)<-c("ID","FirstAmonth.imp")
#For each person, determine first month that imputed D=1
FirstDdat<-data.frame(setDT(dat[dat$D.imputed==1,c("ID","time")])
        [, lapply(.SD, min), by = dat[dat$D.imputed==1,c("ID","time")]$ID])
colnames(FirstDdat)<-c("ID","FirstDmonth.imputed")
#Determine last month of observation for subjects (i.e. censoring month)
CensorTimedat<-data.frame(setDT(dat[c("ID","time")])
        [, lapply(.SD, max), by = dat[,c("ID","time")]$ID])
colnames(CensorTimedat)<-c("ID","maxtime")


#Merge new person-level variables with other time-invariant covariates
#Create person-level dataset
ImputedDat1<-join(FirstAdat,FirstDdat,type="full")
ImputedDat2<-join(ImputedDat1,CensorTimedat,type="full")
cc<-data.table(ImputedDat2,key=c("ID","FirstAmonth.imp"))
dd<-data.table(alldata[,c("ID","sampled","time","A","X","Y",
        "AGE_AT_LAST_VISIT")],key=c("ID","time"))
colnames(dd)[colnames(dd)=="time"]<-"FirstAmonth.imp"
ImputedDat3<-merge(cc,dd)
ff<-data.table(ImputedDat3,key=c("ID","FirstDmonth.imputed"))
gg<-data.table(alldata[,c("ID","time","D")],key=c("ID","time"))
colnames(gg)<-c("ID","FirstDmonth.imputed","D" )

ImputedDat4<-data.frame(gg[ff,])

#Express first A=1 and first D=1 times in terms of years
ImputedDat4$Aage<-(ImputedDat4$FirstAmonth.imp-0)*30.437/365.25
ImputedDat4$Dage<-ImputedDat4$FirstDmonth.imputed*30.437/365.25

#Exclusion criteria
#Exclude if first A=1 after censoring
```

```
        #Exclude if first A=1 after first D=1
        #Exclude if first A=1 before time 0
        d.imp2<-ImputedDat4
        d.imp2$exclude.no.A<-with(d.imp2, ifelse(is.na(FirstAmonth.imp), 1,
                ifelse(Aage>AGE_AT_LAST_VISIT, 1, 0)))
        d.imp2$exclude.priorD<-with(d.imp2, ifelse(!is.na(Dage) &
                !is.na(FirstAmonth.imp) &Dage<Aage, 1, 0))
        d.imp2$exclude.priorA<-with(d.imp2,
                ifelse(is.na(FirstAmonth.imp)==F & Aage<0,1,0))
        d.imp2$exclude<-with(d.imp2,ifelse(exclude.no.A==1|
                exclude.priorD==1 | exclude.priorA, 1, 0))

        d.imp3<-d.imp2[d.imp2$exclude==0,]
        d.imp3$last.age<-d.imp3$AGE_AT_LAST_VISIT+30.437/365.25

        #Create final analysis dataset for export
        #Contains event indicator and time-to-event variables
        d.analysis<-d.imp3
        d.analysis$ade<-with(d.analysis, ifelse(is.na(Dage),0,1))
        d.analysis$fu<-with(d.analysis, ifelse(is.na(Dage),
                last.age-Aage, Dage-Aage))
        d.analysis$fu<-with(d.analysis, ifelse(fu<0,0,fu))
        d.analysis$fu<-round(d.analysis$fu,3)
        #Add exclusion criteria to only look at events after time 0
        d.analysis<-d.analysis[d.analysis$fu>0,]

return(d.analysis)
}
```

4.6.2 Example 3: R code for imputation

In this setting, we specify a joint imputation model for $(\mathcal{A}, \mathcal{D})$ by fitting a logistic regression model for $\mathcal{A}$ conditional on $\mathcal{A}^*$,$\mathcal{D}^*$, X, Y, and m and a logistic regression model for $\mathcal{D}$ conditional on $\mathcal{A}^*$,$\mathcal{D}^*$, X, Y, m, and $\mathcal{A}$ using the subset of 1000 $\times$ 100 audited person-month observations (1000 audited subjects with 100 timepoints each). For each imputation, the imputer accounted for both parameter uncertainty and random noise in the predictions.

```
#Correctly specified imputation models
#Fit two logistic regression models
        modA<-glm(A~time+X+Y+A.star+D.star,data=alldata,
                family="binomial",y=FALSE,model=FALSE)
        modD<-glm(D~A+time+X+Y+A.star+D.star,data=alldata,
                family="binomial",y=FALSE,model=FALSE)


#Create datasets that add imputed values for key values for non-sampled records
#We set up the backbone of those datasets prior to the imputation loop
        dat.A<-data.frame(alldata[,c("ID","sampled",all.vars(formula(modA))[1],
                "Intercept",all.vars(formula(modA))[-1])])
        dat.A<-dat.A[order(dat.A$ID),]
        tempdat.A<-as.matrix(dat.A[,c("Intercept",all.vars(formula(modA))[-1])])

        dat.D<-data.frame(alldata[,c("ID","sampled",all.vars(formula(modD))[1],
                "Intercept",all.vars(formula(modD))[-1])])
        dat.D<-dat.D[order(dat.D$ID),]
################################################################################
#Function to impute values (Same function as in Example 2)
ImputationFn<-function(mod,datty){
        vars<-c("Intercept",all.vars(formula(mod)[-2]))
        newdatmat<-datty[,vars]
        #Account for parameter uncertainty
        #Re-draw from a multivariate normal distribution with mean and variance
        #as the parameter estimates and cov matrix from the regression model.
        desmatrix<-rmvnorm(1, mean=mod$coefficients, sigma=vcov(mod))
        linpred<-as.vector(desmatrix%*%t(newdatmat))

        if (mod$family[1]=="binomial"){ return(linpred)}
        if (mod$family[1]=="gaussian"){
        #Account for random noise
        #Randomly sample from the residuals
                resid.y<-mod$residuals
                imputed.values<-linpred +
                        as.numeric(sample(resid.y,length(linpred),replace=TRUE))
        }
        return(imputed.values)
}
################################################################################
#Impute values
#Impute A
        linpred.A<-ImputationFn(modA,tempdat.A)
```

```
        pred.A<-rbinom(length(linpred.A),1,exp(linpred.A)/(1+exp(linpred.A)))
        dat.A$A[dat.A$sampled==0]<-as.vector(pred.A)[dat.A$sampled==0]
#Impute D
        #Need to update A values here in order to properly impute D status
        dat.D$A[dat.D$sampled==0]<-as.vector(pred.A)[dat.A$sampled==0]
        tempdat.D<-as.matrix(dat.D[,c("Intercept",all.vars(formula(modD))[-1])])

        linpred.D<-ImputationFn(modD,tempdat.D)
        pred.D<-rbinom(length(linpred.D),1,exp(linpred.D)/(1+exp(linpred.D)))
        dat.D$D[dat.D$sampled==0]<-as.vector(pred.D)[dat.D$sampled==0]
```

4.6.3 Example 3: R code for RW component calculations based on imputation model

The imputer needs to calculate and supply two datasets based on the score function of the imputation model, S_{mis} and d. The following code contains several functions. The first two functions evaluate the score function or its derivative for each observation. We note that the unit of observation for imputations was person-months. We, however, need to express this information on the subject-level. Thus, we expand on the functions provided in Example 2 to allow the collapse of information across multiple observations.

We then expand the function to calculate and output the two components (S_{mis} and d) for the RW variance estimator from the imputation model that was provided in the first example. Here, we allow for a joint imputation model and the variable being imputed can be binary or continuous. This function is intended to be used for each imputation.

```
#Function to evaluate score function
#Code here works when imputation model is linear or logistic regression
S_u_function<-function(datty,mod.impute,imputedvar,IDMat){

        imputemod.vars<-c("Intercept",all.vars(formula(mod.impute)[-2]))

if (mod.impute$family[1]=="gaussian"){
        sigma.est<-var(mod.impute$residuals)
        predicted.values<-predict(mod.impute,newdata=datty)
        S_u_1<-(imputedvar-predicted.values)*datty[,imputemod.vars]/sigma.est
        S_sigma_1<-0.5*(-1/sigma.est+ (imputedvar-predicted.values)^2/sigma.est^2)
```

```
        S_u<-cbind(S_u_1,S_sigma_1)
        modelvarcount<-length(imputemod.vars)+1 #Account for sigma
}
if (mod.impute$family[1]=="binomial"){
        #predicted.values is actually predicted LP
        predicted.values<-predict(mod.impute,newdata=datty,returnvar="linpred")
        S_u<-datty[,imputemod.vars]*
                (imputedvar-exp(predicted.values)/(1+exp(predicted.values)))
        modelvarcount<-length(imputemod.vars)
}
        #Need to collapse person-month data to person-level data
        #Take the sum across all observations per individual
        S_up<-data.frame(setDT(S_u)[, lapply(.SD, sum), by = datty$ID])
        colnames(S_up)[1]<-"ID"
        S_upp<-merge(S_up,IDMat,by="ID",all=T)
        S_upp[is.na(S_upp)]<-0
        return(S_upp)

}
#Function to evaluate derivative of score function
#Code here works when imputation model is linear or logistic regression
S2_function<-function(datty,mod.impute){
imputemod.vars<-c("Intercept",all.vars(formula(mod.impute)[-2]))
impvar_n<-length(imputemod.vars)

if (mod.impute$family[1]=="binomial"){
  ImputeX<-datty[,imputemod.vars]
  predicted.values<-predict(mod.impute,newdata=datty,type="link")
  S2_mod<-matrix(NA,ncol(ImputeX),ncol(ImputeX))
  constant.term<-exp(predicted.values)/(1+exp(predicted.values))^2
  for (matrow in 1:ncol(ImputeX)){
          for (matcol in 1:ncol(ImputeX)){
                temp<-(-1)*(ImputeX[,matrow])*((ImputeX[,matcol]))*constant.term
                temp[datty$sampled==0]<-0
        #KEY CHANGE HERE!
        #Note that we divide by the number of subjects, not person-months below
                S2_mod[matrow,matcol]<-sum(temp)/length(unique(datty$ID))
        }
  }
}
return(S2_mod)
}
```

```
#Function that calculates and outputs the required components from imputation model
RW_Components_Imputation_Fn<-function(){
        #Evaluate score function with respect to all parameters in the models
                S_u_mod1_pre<-S_u_function(dat.D,modA,dat.D$A,IDMat)
                S_u_mod1<-S_u_mod1_pre[,!colnames(S_u_mod1_pre)%in%c("ID","Blank")]

                S_u_mod2_pre<-S_u_function(dat.D,modD,dat.D$D,IDMat)
                S_u_mod2<-S_u_mod2_pre[,!colnames(S_u_mod2_pre)%in%c("ID","Blank")]

                S_u<-data.frame(S_u_mod1,S_u_mod2)

                #Preliminary matrix that indicates missingess for each subject
                ImputedMat<-NotImputedMat<-NULL
                ImputedMat1<-matrix(dat.A$sampled[duplicated(dat.A$ID)==F]==0,
                                nrow(S_u_mod1),ncol(S_u_mod1),byrow=FALSE)
                ImputedMat2<-matrix(dat.D$sampled[duplicated(dat.D$ID)==F]==0,
                                nrow(S_u_mod2),ncol(S_u_mod2),byrow=FALSE)
                ImputedMat<-cbind(ImputedMat1,ImputedMat2)
                NOTImputedMat<-1-ImputedMat
#####################################################################
#Calculation of S_mis -- component from imputation model
#Output a dataset that is the evaluated score function for imputed obs
#####################################################################
                S_mis_imp<-S_u*ImputedMat
#####################################################################
#Calculation of D -- component from imputation model
#####################################################################
                S_orig_imp<-S_u*NOTImputedMat
                S2_mod1<-S2_function(dat.A,modA)
                S2_mod2<-S2_function(dat.D,modD)
                n_S2mod1<-ncol(S2_mod1)
                n_S2mod2<-ncol(S2_mod2)
                n_S2mod<-n_S2mod1+n_S2mod2

                S2<-matrix(0,n_S2mod,n_S2mod)
                S2[1:n_S2mod1,1:n_S2mod1]<-S2_mod1
                S2[(n_S2mod1+1):n_S2mod,(n_S2mod1+1):n_S2mod]<-S2_mod2

                #Handle cases with singular matrices
                S3<-try(solve(S2),silent=T)
                S4<-ifelse (is(S3,"try-error"),
                        list((S2+matrix(jitter(rep(.000001,nrow(S2)*ncol(S2))),
                        nrow(S2),ncol(S2)))), list((S2)))
```

```
                S5<-S4[[1]]

                Dmat<-solve(S5)
                d_t<-((-1)*Dmat)%*%t(S_orig_imp)
                d<-t(d_t)
                #Save ordering of IDs to ensure component alignment in later calcs
                IDorder<-S_u_mod1_pre[,c("ID","Blank")]

        return(list(S_mis_imp,d,IDorder))
}
```

4.6.4 Example 3: R code for RW component calculations based on analysis model

The analyst needs to calculate and supply two datasets, u and τ. Similar to previous sections, we provide a function that will calculate and output u and τ for the appropriate analysis models (Kaplan-Meier procedure or Cox regression analysis model) corresponding to this example. These calculations are performed for each imputation.

It is important to highlight that the rigor of calculating u is much greater for our two time-to-event analysis models. This is due to the complexity of the specification and evaluation of estimating equations corresponding to each model. For Cox regression, there are several different ways to parameterize the partial likelihood to be maximized. We likely need to account for ties among event times and have to select the partial likelihood parameterization accordingly. There are several reasonable choices, including the parameterization by Efron, that allow parameter estimation. The challenge in our setting, however, is evaluating the estimating equation for each unique subject rather than each unique event time. Fortunately, there is a built-in function in R that outputs score residuals that we can easily incorporate in our calculations. Meanwhile, the Kaplan-Meier procedure is a non-parametric method and it is not immediately clear how to specify a corresponding estimating equation. Fortunately, previous work by Stute (1995) demonstrated that the Kaplan-Meier procedure can be represented as a sum of random variables that can be used as estimating equations. The technical details go beyond the scope of this paper. The code provided below to evaluate estimating equations at each event time for each observation was based on work by Shepherd et al. (2007).

Fortunately, calculating τ is less difficult for both analysis models. Since we sum across all subjects when calculating τ for cox regression, we no longer are concerned about evaluating for each unique subject. We provide the code using the formula for calculating the information based on Efron's parameterization below. For the Kaplan-Meier procedure, the derivative of the estimating equation corresponding to each event time is 1.

```
#####################################################################
#Analysis components
#####################################################################

#####################################################################
#Function to calculate survival estimate and SE at specific timepoint
#####################################################################
SurvivalEst<-function(stuff){
   mintime<-min(stuff$time)
    #We use the survival estimate from the closest prior time
        #To calculate survival at a given timepoint,
    #We assign a survival estimate of 1 (use max SE here)
    #If no survival estimates at prior timepoints,

        exclude<-sum(times<mintime)
    surv.est<-se.est<-rep(NA,length(times))
    # Only extract estimates if the KM was actually calculated!
    if (stuff$table["events"]!=0 & mintime<=max(times)){
     for (i in (1+exclude):length(times)){
       surv.est[i]<-stuff$surv[stuff$time==max(stuff$time[stuff$time<=times[i]])]
       se.est[i]<-stuff$std.err[stuff$time==max(stuff$time[stuff$time<=times[i]])]
     }
    }
    surv.est<-ifelse(is.na(surv.est),1,surv.est)
    se.est<-ifelse(is.na(se.est),max(stuff$std.err),se.est)

    output<-data.frame(times,surv.est,se.est)
    #If min time > evaltime, return survival est of 1
    return(output)
}

#####################################################################
#Specially formatted KM output to work with Stute estimating eq function
#####################################################################
```

```
myOwnKM <- function(time, delta){
                time<-round(time,6)

            fit <- survfit(Surv(time, delta) ~ 1)
            uniqueAndOrderedTime = round(unique(time)[order(unique(time))],6)
                nevent.giganti<-tapply(delta, time, sum)
                temp_unique<-data.frame(uniqueAndOrderedTime,1)
                temp<-data.frame(fit$time,fit$n.risk,fit$surv)
                temp$fit.time<-round(temp$fit.time,6)
                temp2<-merge(temp_unique,
                                temp[,c("fit.time","fit.n.risk","fit.surv")],
                                by.x="uniqueAndOrderedTime",by.y="fit.time",all=T)
              for (q in 2:nrow(temp2)){
                temp2[q,"fit.n.risk"]<-with(temp2,ifelse(is.na(fit.n.risk[q]),
                                fit.n.risk[q-1],fit.n.risk[q]))
                temp2[q,"fit.surv"]<-with(temp2,ifelse(is.na(fit.surv[q]),
                                fit.surv[q-1],fit.surv[q]))
              }
                  #Add for scenario with only 1 timepoint
                  temp2<-temp2[is.na(temp2$uniqueAndOrderedTime)==F,]
                dataKM = data.frame(time=uniqueAndOrderedTime, delta=NA,
                          nEvents = nevent.giganti, atRisk = temp2$fit.n.risk,
                          KM = temp2$fit.surv, CDF = 1-temp2$fit.surv)
              rownames(dataKM) = uniqueAndOrderedTime
              dataKM = dataKM[as.character(time),]
              dataKM$delta = delta
              #dataKM = dataKM[order(dataKM$time),]
              rownames(dataKM) = 1:nrow(dataKM)

            dataKM
          }
#####################################################################
#Function to evaluate estimating equations from Kaplan Meier based on Stute method
#####################################################################
estimating_equations_KM<-function(myKaplanMeier,reduced=reduced,
        reducedtimes=reducedtimes){
        #Set reduced to 0 for estimates at all times
        #To make run faster and at specific times, specify reducedtimes
        myKaplanMeier<-myKaplanMeier[is.na(myKaplanMeier$time)==FALSE,]

    orderedTime = myKaplanMeier$time[order(myKaplanMeier$time)]
    orderedDelta =  myKaplanMeier$delta[order(myKaplanMeier$time)]
    orderedUniqueKM = unique(myKaplanMeier$KM[order(myKaplanMeier$time)]
```

```
            [orderedDelta == 1])

uniqueTime = unique(myKaplanMeier$time)
orderedUniqueTime = uniqueTime[order(uniqueTime)]
orderedUniqueFailTime = unique(orderedTime[orderedDelta == 1])

N = length(orderedTime)
uniqueN = length(orderedUniqueTime)
uniqueFailureN = length(orderedUniqueFailTime)

orderMatrix = matrix(c(1:N, (1:N)[order(myKaplanMeier$time)]), ncol=2)
colnames(orderMatrix) = c("new", "old")
originalOrder = orderMatrix[order(orderMatrix[,2]), 1]

dresid.dtheta = matrix(0, nrow = uniqueFailureN, ncol = N)
colnames(dresid.dtheta) = orderedTime
rownames(dresid.dtheta) = orderedUniqueFailTime
condition1 = (orderedTime == orderedUniqueFailTime[1]) |
                    ((orderedTime <= orderedUniqueFailTime[1]) &
                            (orderedDelta == 0))
dresid.dtheta[1, (1:N)[condition1]] = 1

phi_ji = matrix(0, nrow = uniqueFailureN, ncol = N)
colnames(phi_ji) = orderedTime
rownames(phi_ji) = orderedUniqueFailTime
indices = (1:N)[orderedTime <= orderedUniqueFailTime[1]]
phi_ji[1, indices] = 1

if (uniqueFailureN>1){
            for (j in 2:uniqueFailureN){
          condition2 = (orderedTime == orderedUniqueFailTime[j] &
                                    (orderedDelta == 1))
          condition3 = ((orderedTime > orderedUniqueFailTime[j-1]) &
                            (orderedTime <= orderedUniqueFailTime[j]) &
                                    (orderedDelta == 0))
          dresid.dtheta[j, (1:N)[condition2 | condition3]] = 1
          dresid.dtheta[j-1, (1:N)[condition2]] = 1
          indices = (1:N)[orderedTime <= orderedUniqueFailTime[j]]
          phi_ji[j, indices] = 1
        }
    }
condition4 = (orderedTime > orderedUniqueFailTime[uniqueFailureN] &
                                    (orderedDelta == 0))
```

```
dresid.dtheta[uniqueFailureN, (1:N)[condition4]] = 1
orderedDeltaMatr = matrix(orderedDelta, nrow = uniqueFailureN,
                                          ncol = N, byrow=TRUE)

  H = H0 = H1 = rep(NA, N)
  for (i in 1:N){
    H[i] = mean(orderedTime <= orderedTime[i])
    H0[i] = mean((orderedTime <= orderedTime[i])*(1 - orderedDelta))
    H1[i] = mean((orderedTime <= orderedTime[i])*orderedDelta)
  }

H0dv = diff(c(0, H0))
H1dw = diff(c(0, H1))
H1dwPerY = H0dvPerY = rep(0, N)
H0dvPerY[orderedTime * (1 - orderedDelta) == orderedTime] =
            H0dv[orderedTime * (1 - orderedDelta) == orderedTime]
H1dwPerY[orderedTime * orderedDelta == orderedTime] =
            H1dw[orderedTime * orderedDelta == orderedTime]

################ making sure that 1-H is never zero
Hadj = H
Hadj[Hadj == 1] = .99

multiplier = 1/(1 - Hadj)
multiplier[H==1] = 0   ### this likely fixes the the problem of division by 0

gamma0 = exp(cumsum(c(0, (H0dvPerY * multiplier)  )))[1:N]
Vji = gamma_j2 = gamma_j1 = phi_ji*0
vValue = wValue  = orderedTime

    if (reduced==1){
      #Extra code to get closest to evaltime without going over
      dif<-as.numeric(rownames(phi_ji))-evaltime
      switchtime<-max(rownames(phi_ji)[dif<=0])
      reducedtimes<-unique(c(min(rownames(phi_ji)),reducedtimes,switchtime))
      #Only evaluate at specific timepoints
      orderedUniqueKM<-orderedUniqueKM[rownames(phi_ji)%in%reducedtimes]
      orderedUniqueFailTime<-orderedUniqueFailTime[
            rownames(phi_ji)%in%reducedtimes]
      orderedUniqueFailTime[orderedUniqueFailTime==switchtime]<-evaltime
      uniqueFailureN<-length(orderedUniqueFailTime)
      phi_ji<-phi_ji[rownames(phi_ji)%in%reducedtimes,]
      gamma_j1<-gamma_j1[rownames(gamma_j1)%in%reducedtimes,]
```

```
          gamma_j2<-gamma_j2[rownames(gamma_j2)%in%reducedtimes,]
          Vji<-Vji[rownames(Vji)%in%reducedtimes,]
        }
    for(j in 1:nrow(gamma_j1)){
      for(i in 1:ncol(gamma_j1)){
        indicatorForGamma1 = as.numeric((orderedTime[i] < wValue) & phi_ji[j,])
        gamma_j1[j,i] = multiplier[i]*sum(indicatorForGamma1*gamma0*H1dwPerY)
      }
    }

    for(j in 1:nrow(gamma_j1)){
      for(i in 1:ncol(gamma_j1)){
        indicatorForGamma2 = as.numeric((vValue < orderedTime[i]))
        gamma_j2[j,i]=sum(multiplier*indicatorForGamma2*gamma_j1[j,]*H0dvPerY)
        Vji[j, i] = phi_ji[j, i] * gamma0[i] * orderedDelta[i] +
                                  gamma_j1[j, i]*(1-orderedDelta[i])-gamma_j2[j,i]
      }
    }
    res=Vji[,originalOrder]-
                        (1-matrix(orderedUniqueKM,nrow=uniqueFailureN,ncol= N))

        rownames(res)<-orderedUniqueFailTime
    list(dl.dtheta = res, dresid.dtheta = dresid.dtheta[, originalOrder])
  }

#####################################################################
#Calculation of u
#Evaluate the estimating equation for each observation
#####################################################################
U_function<-function(dat.imputed,outcome,mod.analysis=NULL,IDorder=IDorder,
        reduced=0,reducedtimes=c(0,1)){
#This function becomes computationally expensive
#May want to reduce the number of timepoints evaluated
#reduced, reducedtimes, is for estimating_equations_KM function
#Set to 'no' as default
if (outcome%in%c("SurvEst","CoxPH")){
        survivaldat.imputed<-dat.imputed
        timevar<-survivaldat.imputed$fu
        eventvar<-survivaldat.imputed$ade
}
if (outcome=="SurvEst"){
        #Evaluate KM estimates to use in estimating eq function
        tmpKM = myOwnKM(timevar, eventvar)
```

```
        #Estimating Eq from Stute
        #Note that KM estimating equations are set to 0 before any events occur
        bundle1=estimating_equations_KM(myKaplanMeier=tmpKM,reduced=reduced,
                                reducedtimes=reducedtimes)$dl.dtheta

        #Loop through timepoints of interest (times) to assign values to U matrix
        #Estimating equations were evaluated at all event times in fn above
        #Check to see if estimating equation evaluated at that timepoint
        #If yes, assign it to row of matrix corresponding to that timepoint
        #If not, assign value of zero
        U_imp<-matrix(0,length(times),nrow(dat.imputed))
        totalrows<-as.numeric(rownames(bundle1))
        rowcount<-0

        for (q in times){
                extract<-max(c(totalrows[abs(totalrows-q)<0.0001],"-Inf"))
                rowcount<-rowcount+1
                if (extract!="-Inf") {
                 U_imp[rowcount,]<-bundle1[rownames(bundle1)==extract,]}
        }
    U_imp<-t(U_imp)
        colnames(U_imp)<-paste("U_",1:ncol(U_imp),sep="")
        rownames(U_imp)<-NULL
}
if (outcome=="CoxPH"){
        analy.vars<-names(mod.analysis$coef)
        analy.vars<-ifelse(analy.vars=="(Intercept)","Intercept",analy.vars)

        if (length(analy.vars)==1){
                #Use score residuals from R output
                S_imp_wide<-residuals(mod.analysis,type="score")
                U_imp<-matrix(S_imp_wide,length(S_imp_wide),1)
                colnames(U_imp)<-paste("U_",analy.vars,sep="")
        }
        if (length(analy.vars)!=1){
                S_imp_wide<-residuals(mod.analysis,type="score")
                U_imp<-S_imp_wide
                colnames(U_imp)<-paste("U_",analy.vars,sep="")

        }
}
```

```
#Have rows for each obs in the imputation dataset
#As opposed to just obs for those in the analysis dataset
#Sync the row order of obs here with row order in imputation components
aa<-data.frame(as.character(dat.imputed$ID),U_imp,stringsAsFactors=FALSE)
colnames(aa)<-c("ID",colnames(U_imp))
bb<-merge(IDorder,aa,by="ID",all=T)
U_imp_all<-bb[,colnames(U_imp)]
U_imp_all[is.na(U_imp_all)]<-0

return(U_imp_all)
}

#####################################
#Function for calculating information
#Based on Efron method of handling ties
#####################################
InfoFn<-function(modcoef,time,event,dat,variables){
#Number of unique, ordered death times
sorted.death.times<-sort(unique(time[event==1]))
Ndeathtimes<-length(sorted.death.times)

I<-matrix(0,length(variables),length(variables))
for (p in 1:length(variables)){
        varA<-variables[p]
        for (q in 1:length(variables)){
        SecondDeriv<-0
        varB<-variables[q]

        #Chunks for Infomation
        #Essentially, the information consists of multiple components
        #First chunk designated by A (11 corresponds to 1st numerator/component)
        #Overall chunk is labeled chunk
        #Need to iterate through unique death times (Ndeathtimes),
        # and then within each event at a given death time (Nevents)
        #Need to distinguish between observations with events at timepoint
        #and observations at risk at timepoint(.atrisk)
        for (i in 1:Ndeathtimes){
                N.AtRisk<-nrow(dat[time>=sorted.death.times[i],])
                Nevents<-nrow(dat[time==sorted.death.times[i] & event==1,])

                dat_AtRisk<-as.matrix(dat[time>=sorted.death.times[i],variables])
                LP_AtRisk<-exp(dat_AtRisk%*%t(t(modcoef)))
```

```
                dat_Events<-as.matrix(dat[time==sorted.death.times[i] & event==1,
                                          variables])
                LP_Events<-exp(dat_Events%*%t(t(modcoef)))

                colnames(dat_Events)<-colnames(dat_AtRisk)<-variables

                Chunk.A11<-sum(dat_AtRisk[,varA]*dat_AtRisk[,varB]*
                                          LP_AtRisk)
                Chunk.A21<-sum(LP_AtRisk)

                Chunk.B11<-sum(dat_AtRisk[,varA]*LP_AtRisk)
                Chunk.B13<-sum(dat_AtRisk[,varB]*LP_AtRisk)
                ChunkA<-ChunkB<-0

                for (j in 1:Nevents){
                        Chunk.A12<-(j-1)/Nevents*sum(dat_Events[,varA]*
                                dat_Events[,varB]*LP_Events)
                        Chunk.A22<-(j-1)/Nevents*sum(LP_Events)
                        ChunkA_temp<- - (Chunk.A11-Chunk.A12)/(Chunk.A21-Chunk.A22)
                        ChunkA<-ChunkA_temp+ChunkA

                        Chunk.B12<-(j-1)/Nevents*sum(dat_Events[,varA]*LP_Events)
                        Chunk.B14<-(j-1)/Nevents*sum(dat_Events[,varB]*LP_Events)

                        ChunkB_temp<- (Chunk.B11-Chunk.B12)*(Chunk.B13-Chunk.B14)/
                                        (Chunk.A21-Chunk.A22)^2
                        ChunkB<-ChunkB_temp+ChunkB
                Chunk<-ChunkA+ChunkB

                }
        SecondDeriv<-SecondDeriv+Chunk
        }
        I[p,q]<-(SecondDeriv)
}
}
return(I)
}

RW_Components_Analysis_Fn<-function(Analysis){
if (Analysis=="CoxPH"){
        #U#
        U_imp_cox<-U_function(d.analysis,outcome=Analysis,
                mod.analysis=mod.analysis,IDorder=IDorder)
```

```
        #Tau#
        tempdat<-d.analysis #Prettify code below with shorter dat name
        tau_imp_cox<-(-1/n)*InfoFn(mod.analysis$coef,time=tempdat$fu,
                                   event=tempdat$ade,dat=tempdat,variables=analy.vars)
        return(list(U_imp_cox,tau_imp_cox))

}
if (Analysis=="SurvEst"){
        #U#
        #Takes a long time computationally!
        U_imp_surv<-U_function(d.analysis,outcome=Analysis,mod.analysis=NULL,
                IDorder=IDorder,reduced=0,reducedtimes=c(times[1],evaltime))

        #Tau#
                if (is.null(ncol(U_imp_surv))==FALSE){
                        tau_imp_surv<-diag(nrow(d.analysis)/n,ncol(U_imp_surv))}
                if (is.null(ncol(U_imp_surv))){
                        tau_imp_surv<-diag(nrow(d.analysis)/n,1)}
        }
        return(list(U_imp_surv,tau_imp_surv))
}
```

4.6.5 Example 3: R code for RW multiple imputation variance calculation

The above components are calculated using one imputed dataset. The following code provides an outline for calculating the variance estimator across 20 imputations for both the Cox regression parameter estimate and the Kaplan-Meier estimated incidence at 60 months.

```
#Function to calculate RW based on calculated components
RobinsWangFn<-function(U_imp_sum,kappa_sum,tau_sum,d,Analysis){
        u_bar<-t(t(U_imp_sum/Nimpute))
        u_bar[u_bar=="NaN"]=0

        kappa<-t(t(kappa_sum/(n*Nimpute)))
        alpha<-(t(d)%*%d)/n
        omega<-(t(u_bar)%*%t(t(u_bar)))/n
        delta<-omega+kappa%*%alpha%*%t(kappa) +
                        (1/n)*(kappa%*%t(d)%*%u_bar + t(kappa%*%t(d)%*%u_bar))
        tau<-tau_sum/(Nimpute)
        GAMMA<-(1/n)*t(t(solve(tau)))%*%delta%*%t(solve(tau))
}
```

```
#Initialize select variables
    S_mis<-NULL
    U_imp_sum_cox<-U_imp_sum_surv<-0
    kappa_sum_cox<-kappa_sum_surv<-0
    tau_sum_cox<-tau_sum_surv<-0
    surv.est<-surv.se<-matrix(NA,length(times),Nimpute)
    cox.est<-cox.se<-matrix(NA,length(CoxVar),Nimpute)

for (p in 1:Nimpute) {
        ##############################################
        #Replace missing values with imputed value
        ##############################################
        #A
        linpred.A<-ImputationFn(modA,tempdat.A)
        pred.A<-rbinom(length(linpred.A),1,exp(linpred.A)/(1+exp(linpred.A)))
        dat.A$A[dat.A$sampled==0]<-as.vector(pred.A)[dat.A$sampled==0]
        ##############################################
        ##D
        ##############################################
        #Need to update A values here in order to properly impute D
        dat.D$A[dat.D$sampled==0]<-as.vector(pred.A)[dat.A$sampled==0]
        tempdat.D<-as.matrix(dat.D[,c("Intercept",all.vars(formula(modD))[-1])])
        linpred.D<-ImputationFn(modD,tempdat.D)
        pred.D<-rbinom(length(linpred.D),1,exp(linpred.D)/(1+exp(linpred.D)))
        dat.D$D[dat.D$sampled==0]<-as.vector(pred.D)[dat.D$sampled==0]

        ##############################################
        #Undiscretize data to create analysis dataset
        ##############################################
        aaa<-data.table(data.frame(alldata[,c("ID","sampled","time","D")],pred.D),
                key=c("ID","time"))
        bbb<-data.table(data.frame(alldata[,c("ID","time","A")],pred.A),
                key=c("ID","time"))
        dat<-data.frame(aaa[bbb,])
        d.analysis<-BuildAnalysisDat(dat)

        ##############################################
        #Perform analyses
        ##############################################
        #Cox regression
        mod.analysis<-coxph(Surv(fu,ade)~X,data=d.analysis,y=FALSE)
        analy.vars<-all.vars(formula(mod.analysis)[-2])
```

```
#Kaplan-Meier analysis
KM.tdmiformatted<-SurvivalEst(with(d.analysis,
        summary(survfit(Surv(fu,ade)~1))))

#Capture estimates and corresponding SE from each imputation
surv.est[,p]<-KM.tdmiformatted[,"surv.est"]
surv.se[,p]<-KM.tdmiformatted[,"se.est"]
cox.est[,p]<-mod.analysis$coef[CoxVar]
cox.se[,p]<-sqrt(diag(vcov(mod.analysis))[CoxVar])

#Create matrix containing IDs - helpful for merging + ordering matrices
IDMat<-unique(data.frame(alldata[,"ID"],1,stringsAsFactors=FALSE))
colnames(IDMat)<-c("ID","Blank")

#####################################################################
#Evaluate all components for a given imputation
#####################################################################
ImputationComponents<-RW_Components_Imputation_Fn()
S_mis_imp<-ImputationComponents[[1]]
d<-ImputationComponents[[2]] #Only need to calculate d once
IDorder<-ImputationComponents[[3]]

AnalysisComponents_cox<-RW_Components_Analysis_Fn(Analysis="CoxPH")
AnalysisComponents_surv<-RW_Components_Analysis_Fn(Analysis="SurvEst")
U_imp_cox<-AnalysisComponents_cox[[1]]
U_imp_surv<-AnalysisComponents_surv[[1]]
U_imp_cox[is.na(U_imp_cox)]<-0
U_imp_surv[is.na(U_imp_surv)]<-0
tau_imp_cox<-AnalysisComponents_cox[[2]]
tau_imp_surv<-AnalysisComponents_surv[[2]]


#Create summations of certain components across imputations
U_imp_sum_cox<-U_imp_sum_cox+U_imp_cox
tau_sum_cox<-tau_sum_cox+tau_imp_cox
U_imp_sum_surv<-U_imp_sum_surv+U_imp_surv
tau_sum_surv<-tau_sum_surv+tau_imp_surv

#Calculate kappa for a given imputation
kappa_imp_cox<-t(U_imp_cox)%*%t(t(S_mis_imp))
kappa_sum_cox<-kappa_sum_cox+kappa_imp_cox
kappa_imp_surv<-t(U_imp_surv)%*%t(t(S_mis_imp))
kappa_sum_surv<-kappa_sum_surv+kappa_imp_surv
```

```
        ##############################################
} #This bracket ends imputation portion of loop

#####################################################################
#Combine components together to calculate RW variance estimator
#####################################################################
        TDMIest.surv<-apply(surv.est,1,mean)
        TDMIest.cox<-mean(cox.est)

        GAMMA.cox<-RobinsWangFn(U_imp_sum_cox,kappa_sum_cox,tau_sum_cox,d,
                Analysis="CoxPH")
        GAMMA.surv<-RobinsWangFn(U_imp_sum_surv,kappa_sum_surv,tau_sum_surv,d,
                Analysis="SurvEst")

        #Output RW variance estimates
        RobinsWangSE.surv<-diag(sqrt(GAMMA.surv))
        RobinsWangSE.cox<-diag(sqrt(GAMMA.cox))[1]

        #Compare to SE estimates from Rubin
        Rubin.se.surv<-sqrt(apply(surv.se^2,1,mean)+(Nimpute+1)*
                apply(surv.est,1,var)/Nimpute)
        Rubin.se.cox<-sqrt(mean(cox.se^2)+(Nimpute+1)*
                var(as.vector(cox.est))/Nimpute)

        #Output estimates
        TDMIest.surv[times==evaltime]

## [1] 0.8439519

        TDMIest.cox

## [1] -1.70452

        #Output variance estimates for RW and Rubin
        #Incidence at 5 years
        RobinsWangSE.surv[times==evaltime]^2

## [1] 8.119119e-05

        Rubin.se.surv[times==evaltime]^2

## [1] 0.0001474951

        #Cox variance estimate
        RobinsWangSE.cox^2
```

```
## [1] 0.005370765
```

```
        Rubin.se.cox^2
```

```
## [1] 0.01453294
```

4.6.6 Example 3: Results

For this single simulation, the RW imputation standard error estimate is smaller than the RR imputation standard error estimate corresponding to both the Cox regression parameter estimate (0.0733 vs. 0.1206) and the estimated event incidence at 60 months (0.0090 vs. 0.0121).

When we repeated this simulation 1000 times, the results were similar. With respect to the Cox regression parameter estimate, the mean RW imputation standard error estimate (0.0100) was similar to the empirical standard error estimate (0.0099) and smaller than the RR imputation standard error estimate (0.0138). The imputation variance estimator proposed by Robins and Wang had better coverage (0.94) compared to Rubin's approach (0.99).

The mean RW imputation standard error estimate corresponding to the Cox regression parameter estimate (0.0739) was also smaller than the RR imputation standard error estimate (0.1192). For reference, the empirical standard error estimate was 0.0821. The estimated coverage using the RW standard error estimate was near nominal levels (0.93), while the coverage using the RR standard error estimate (0.99) was inflated.

4.7 Discussion

To our knowledge, this is the first instance of statistical code being shared publicly for the calculation of Robins and Wang's imputation variance estimator. As the popularity of multiple imputation grows and its implementation is increasingly performed by those with non-statistical backgrounds, it is important that analysts are familiar with and able to carry out best practices regarding variance calculations. We have highlighted several situations where the standard variance estimation based on Rubin's rules is biased. The annotated R code provided in this manuscript demonstrates how to calculate Robins and Wang's imputation variance estimates across multiple settings to obtain estimates that are unbiased in large samples. We hope that this work helps address the biggest drawback of Robins and Wang's approach - its complexity - and reduces the barriers to implementation when relevant.

Our examples were specifically chosen to illustrate implementation for various imputation and analysis models. We outlined how to perform necessary calculations for multiple variable imputation for continuous or binary variables. We also chose four unique analysis models (linear regression, logistic regression, Cox regression, and Kaplan-Meier estimation) to demonstrate implementation in settings that researchers are likely to encounter. Although an earlier paper illustrated the improvement of the RW variance estimator relative to Rubin's in a simple example involving linear regression imputation and analysis models, this paper did not provide analysis code (Hughes et al., 2016); our first example provides the code for HST's example. In addition, we felt it was important to show how this technique could be used in other settings, particularly those involving time-to-event analyses.

The three examples provided above also highlight several incompatibility scenarios that we feel are relevant to a practical contemporary observational data analysis: exclusion based on an invariant, non-imputed value (i.e., static exclusion), exclusion based on an imputed value (i.e., non-static exclusion), and time-discretized data where the unit of observation is different between imputation and analysis models. In all three examples, the bias of the popular estimate for variance estimation was substantial, resulting in very conservative confidence intervals.

We note that the calculation of the RW variance estimator for the last two examples involved fully parametric imputation models. This was reasonable because the missingness structure in our examples was simple: a subset of subjects had complete data and the remaining subjects were missing select variables. In EHR datasets, different subjects will often be missing different combinations of values. In practice, we often use more flexible approaches such as fully conditional specified methods like

multiple imputation by chained equations (MICE). Future work should investigate how this method performs when imputation models like MICE are used.

While the R code provided in this manuscript will hopefully be a useful tool for future implementors, we acknowledge that there are limits to its generalizability. Future implementors hoping to impute more than two variables or using imputation models other than logistic or linear regression will need to make modifications to our code. While we selected four popular analysis models, we realize future implementors may be interested in other models. We encourage researchers to build off the existing code provided here to create functions generalizable to more settings. We are aware of at least one group that has proposed to construct an R package for calculating RW imputation variance estimates, but to our knowledge the work was never finished (Reilly, 2009). Creating software that generalizes RW would be a challenging, but worthy, endeavor.

4.8 Appendix C

Table 4.1: Estimated coverage probabilities when imputation variance estimated using Rubin's approach for different validation subsample sizes (n= 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000) as well as different inclusion thresholds ($\mathcal{A} > \{-\infty$,-1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3$\}$).

	500	1000	1500	2000	2500	3000	3500	4000
$\mathcal{A} > 3$	1.00	1.00	1.00	1.00	1.00	0.99	0.97	0.96
$\mathcal{A} > 2.5$	1.00	1.00	1.00	0.99	0.99	0.98	0.97	0.95
$\mathcal{A} > 2$	0.99	1.00	1.00	0.99	0.99	0.98	0.97	0.95
$\mathcal{A} > 1.5$	0.98	0.99	0.99	0.99	0.99	0.98	0.97	0.96
$\mathcal{A} > 1$	0.97	0.98	0.98	0.98	0.98	0.98	0.97	0.96
$\mathcal{A} > 0.5$	0.96	0.97	0.97	0.97	0.97	0.97	0.96	0.95
$\mathcal{A} > 0$	0.95	0.96	0.97	0.96	0.97	0.96	0.96	0.95
$\mathcal{A} >$ -0.5	0.94	0.95	0.96	0.96	0.96	0.96	0.96	0.95
$\mathcal{A} >$ -1	0.94	0.95	0.96	0.96	0.96	0.96	0.96	0.95
$\mathcal{A} > -\infty$	0.94	0.95	0.95	0.95	0.96	0.96	0.95	0.94

Table 4.2: Estimated coverage probabilities when imputation variance estimated using Robins and Wang's approach for different validation subsample sizes (n= 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000) as well as different inclusion thresholds ($\mathcal{A} > \{-\infty$,-1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5, 3$\}$).

	500	1000	1500	2000	2500	3000	3500	4000
$\mathcal{A} > 3$	0.96	0.97	0.96	0.96	0.96	0.95	0.95	0.95
$\mathcal{A} > 2.5$	0.95	0.96	0.95	0.95	0.95	0.95	0.95	0.95
$\mathcal{A} > 2$	0.96	0.96	0.95	0.95	0.95	0.95	0.95	0.95
$\mathcal{A} > 1.5$	0.96	0.95	0.95	0.95	0.95	0.95	0.95	0.96
$\mathcal{A} > 1$	0.96	0.96	0.96	0.95	0.95	0.95	0.95	0.96
$\mathcal{A} > 0.5$	0.96	0.96	0.96	0.96	0.95	0.96	0.95	0.95
$\mathcal{A} > 0$	0.97	0.96	0.96	0.95	0.95	0.95	0.95	0.95
$\mathcal{A} >$ -0.5	0.97	0.96	0.96	0.95	0.95	0.95	0.95	0.94
$\mathcal{A} >$ -1	0.97	0.97	0.96	0.96	0.95	0.95	0.95	0.94
$\mathcal{A} > -\infty$	0.97	0.97	0.96	0.95	0.95	0.95	0.95	0.94

CHAPTER 5

CONCLUSION

The practical objective behind the research presented in this dissertation was to advance the methodological sophistication regarding the evaluation and analysis of audit data. This research was motivated by actual discussions that occurred as part of applied collaborations with Caribbean, Central and South America network for HIV epidemiology (CCASAnet), a consortium of research groups from seven Latin American countries that collect and share HIV care and treatment data. With a strong commitment to data quality, CCASAnet has been routinely performing source data verification since its inception. Our post-audit strategy has typically been based on subjective interpretation of error rates. If the error-rate for a given variable is considered high at a site, it is recommended that all entries for that variable be reviewed. If the error-rate is considered acceptable, we proceed with analyses. After several years, we started asking whether we could do better. Could we incorporate statistical methods to better evaluate the audit and improve subsequent analyses? The work presented as part of this dissertation demonstrates that we can incorporate statistical methods to better utilize data collected as part of data audits.

The first task with respect to evaluating data audits was to justify their continued implementation. In Chapter 2, we proposed a framework to assess data audit impact beyond simple error rate calculation. The findings presented in this chapter show the impact that data errors (as discovered in an audit) have on analysis results. Furthermore, the results presented in this chapter show that data can also change post-audit, in part motivated by the audit. Together, these comparisons illustrate that data audits can have an impact on how study findings are clinically interpreted as a result of improved data quality, especially for variables that had not previously been audited. We encourage future investigators to use a similar framework that incorporates both error rate calculation and fitting analysis models using pre-audit and audited records when evaluating their data audits.

Once we identified that the originally collected data were error-prone and likely to influence our statistical inferences, it became important to identify a way to efficiently adjust our analysis estimates without having to audit the entire dataset. The time-discretized multiple imputation model proposed in Chapter 3 demonstrated that one can use statistical methods to combine error-prone data for all records with validated data for a subset of records and obtain unbiased estimates. Beyond the data audit

setting, we note that this work was the first to our knowledge to simultaneously address errors in predictors, censored failure times, event indicators, and inclusion criteria in a time-to-event analysis.

While we were able to obtain approximately unbiased estimates using this TDMI approach, we were not able to calculate imputation variance estimates using Rubin's approach as it assumes compatibility between the imputation and analysis models. Instead, we selected an alternative imputation variance estimator proposed by Robins and Wang (2000) when calculating the 95% confidence intervals reported in Chapter 4. We acknowledge, however, that this approach was challenging to implement due its complexity and the paucity of other implementations in the literature. The work presented in Chapter 4 was intended to make this approach more accessible to contemporary multiple imputation implementors who may be discouraged by the technical complexity of the original manuscript. This chapter provides a tutorial with annotated R code using several examples with common scenarios regarding imputation models, analysis models, and incompatibility due to discretization or inclusion criteria that one may encounter when analyzing datasets. We were encouraged that our coverage estimates were closer to nominal levels in each of our examples and encourage future implementors to use this variance estimator, especially when the imputation and analysis models are incompatible.

The work contained in this dissertation provides an overview for using statistical methods to improve the data audit process, from understanding the propensity for errors in an observational dataset to improving analyses by efficiently accounting for those same errors. Our findings demonstrate that statistical methods can be incorporated to better utilize audit data, but we acknowledge there are still many exciting opportunities to extend our work further. While the work in this dissertation was based on a random sample for selecting the audit subgroup, alternative sampling designs, including oversampling exposures or events considered more likely to be error-prone, might be optimal. We realize future implementors may be interested in models other than those highlighted in the examples calculating the Robins and Wang variance estimator. Investigations regarding its implementation in additional settings when there is incompatibility between imputation and analysis models such as propensity score analyses or with alternative imputation approaches such as multiple imputation by chained equations would also be beneficial. We encourage researchers to extend the code provided as part of this dissertation to create functions generalizable to more settings.

REFERENCES

Chan, K. S., Fowles, J. B. and Weiner, J. P. (2010), Electronic health records and reliability and validity of quality measures: a review of the literature, *Medical Care Research and Review* **67**(5), 742–752.

Cole, S. R., Chu, H. and Greenland, S. (2006), Multiple-imputation for measurement-error correction, *International Journal of Epidemiology* **35**(4), 1074–1081.

Cook, J. R. and Stefanski, L. A. (1994), Simulation-extrapolation estimation in parametric measurement error models, *Journal of the American Statistical Association* **89**(428), 1314–1328.

D'Agostino, R. B., Lee, M.-L., Belanger, A. J., Cupples, L. A., Anderson, K. and Kannel, W. B. (1990), Relation of pooled logistic regression to time dependent Cox regression analysis: the Framingham Heart Study, *Statistics in Medicine* **9**(12), 1501–1515.

Duda, S. N., Shepherd, B. E., Gadd, C. S., Masys, D. R. and McGowan, C. C. (2012), Measuring the quality of observational study data in an international HIV research network, *PloS One* **7**(4), e33908.

Edwards, J. K., Cole, S. R., Troester, M. A. and Richardson, D. B. (2013), Accounting for misclassified outcomes in binary regression models using multiple imputation with internal validation data, *American Journal of Epidemiology* **177**(9), 904–912.

Floyd, J. S., Heckbert, S. R., Weiss, N. S., Carrell, D. S. and Psaty, B. M. (2012), Use of administrative data to estimate the incidence of statin-related rhabdomyolysis, *Journal of the American Medical Association* **307**(15), 1580–1582.

Giganti, M. J., Luz, P. M., Caro-Vega, Y., Cesar, C., Padgett, D., Koenig, S., Echevarria, J., McGowan, C. C. and Shepherd, B. E. (2015), A comparison of seven Cox regression-based models to account for heterogeneity across multiple HIV treatment cohorts in Latin America and the Caribbean, *AIDS Research and Human Retroviruses* **31**(5), 496–503.

Hernán, M. A., Brumback, B. and Robins, J. M. (2001), Marginal structural models to estimate the joint causal effect of nonrandomized treatments, *Journal of the American Statistical Association* **96**(454), 440–448.

Houston, L., Probst, Y. and Humphries, A. (2015), Measuring data quality through a source data verification audit in a clinical research setting, *Stud Health Technol Inform* **214**, 107–13.

Huang, Y. and Wang, C. (2000), Cox regression with accurate covariates unascertainable: A nonparametric-correction approach, *Journal of the American Statistical Association* **95**(452), 1209–1219.

Hughes, R., Sterne, J. and Tilling, K. (2016), Comparison of imputation variance estimators, *Statistical methods in medical research* **25**(6), 2541–2557.

Hunsberger, S., Albert, P. S. and Dodd, L. (2010), Analysis of progression-free survival data using a discrete time survival model that incorporates measurements with and without diagnostic error, *Clinical Trials* **7**(6), 634–642.

Kiragga, A. N., Castelnuovo, B., Schaefer, P., Muwonge, T. and Easterbrook, P. J. (2011), Quality of data collection in a large HIV observational clinic database in sub-saharan africa: implications for clinical research and audit of care, *Journal of the International AIDS Society* **14**(1), 3.

Korn, E. L., Dodd, L. E. and Freidlin, B. (2010), Measurement error in the timing of events: effect on survival analyses in randomized clinical trials, *Clinical Trials* **7**(6), 626–633.

Li, Y. and Lin, X. (2003), Functional inference in frailty measurement error models for clustered survival data using the SIMEX approach, *Journal of the American Statistical Association* **98**(461), 191–203.

Little, R. J. and Rubin, D. B. (2014), *Statistical analysis with missing data*, Vol. 333, John Wiley & Sons.

Lumley, T., Shaw, P. A. and Dai, J. Y. (2011), Connections between survey calibration estimators and semiparametric models for incomplete data, *International Statistical Review* **79**(2), 200–220.

Magaret, A. S. (2008), Incorporating validation subsets into discrete proportional hazards models for mismeasured outcomes, *Statistics in Medicine* **27**(26), 5456–5470.

McClintock, B. T., Johnson, D. S., Hooten, M. B., Ver Hoef, J. M. and Morales, J. M. (2014), When to be discrete: the importance of time formulation in understanding animal movement, *Movement Ecology* **2**(1), 1.

McGowan, C. C., Cahn, P., Gotuzzo, E., Padgett, D., Pape, J. W., Wolff, M., Schechter, M. and Masys, D. R. (2007), Cohort profile: Caribbean, Central and South America Network for HIV research (CCASAnet) collaboration within the international Epidemiologic databases to evaluate AIDS (IeDEA) programme, *International Journal of Epidemiology* **36**(5), 969–976.

Mitchel, J. T., Kim, Y. J., Choi, J., Park, G., Cappi, S., Horn, D., Kist, M. and D'Agostino Jr, R. B. (2011), Evaluation of data entry errors and data changes to an electronic data capture clinical trial database, *Drug Information Journal* **45**(4), 421–430.

Muthee, V., Bochner, A. F., Osterman, A., Liku, N., Akhwale, W., Kwach, J., Prachi, M., Wamicwe, J., Odhiambo, J., Onyango, F. et al. (2018), The impact of routine data quality assessments on electronic medical record data quality in Kenya, *PloS One* **13**(4), e0195362.

Nakamura, T. (1990), Corrected score function for errors-in-variables models: Methodology and application to generalized linear models, *Biometrika* **77**(1), 127–137.

Nicol, E., Dudley, L. and Bradshaw, D. (2016), Assessing the quality of routine data for the prevention of mother-to-child transmission of HIV: An analytical observational study in two health districts with high HIV prevalence in South Africa, *International journal of medical informatics* **95**, 60–70.

Oh, E. J., Shepherd, B. E., Lumley, T. and Shaw, P. A. (2018), Considerations for analysis of time-to-event outcomes measured with error: bias and correction with SIMEX, *Statistics in Medicine* **37**(8), 1276–1289.

Prentice, R. (1982), Covariate measurement errors and parameter estimation in a failure time regression model, *Biometrika* **69**(2), 331–342.

Puttkammer, N., Baseman, J. G., Devine, E. B., Valles, J., Hyppolite, N., Garilus, F., Honoré, J.-G., Matheson, A. I., Zeliadt, S., Yuhas, K. et al. (2016), An assessment of data quality in a multi-site electronic medical record system in Haiti, *International Journal of Medical Informatics* **86**, 104–116.

Reilly, J. (2009), Unbiased variance estimates for multiple imputation in R., *in* 'The R User Conference 2009. Rennes, France'.

Richardson, B. A. and Hughes, J. P. (2000), Product limit estimation for infectious disease data when the diagnostic test for the outcome is measured with uncertainty, *Biostatistics* **1**(3), 341–354.

Robins, J. M. and Wang, N. (2000), Inference for imputation estimators, *Biometrika* **87**(1), 113–124.

Schafer, J. L. (1999), Multiple imputation: a primer, *Statistical Methods in Medical Research* **8**(1), 3–15.

Shepherd, B. E., Gilbert, P. B. and Lumley, T. (2007), Sensitivity analyses comparing time-to-event outcomes existing only in a subset selected postrandomization, *Journal of the American Statistical Association* **102**(478), 573–582.

Shepherd, B. E. and Rebeiro, P. F. (2017), Assessing and interpreting the association between continuous covariates and outcomes in observational studies of HIV using splines, *Journal of Acquired Immune Deficiency Syndromes (1999)* **74**(3), e60.

Shepherd, B. E., Shaw, P. A. and Dodd, L. E. (2012), Using audit information to adjust parameter estimates for data errors in clinical trials, *Clinical Trials* **9**(6), 721–729.

Shepherd, B. E. and Yu, C. (2011), Accounting for data errors discovered from an audit in multiple linear regression, *Biometrics* **67**(3), 1083–1091.

Skinner, C. J. and Humphreys, K. (1999), Weibull regression for lifetimes measured with error, *Lifetime Data Analysis* **5**(1), 23–37.

Smith, C. T., Stocken, D. D., Dunn, J., Cox, T., Ghaneh, P., Cunningham, D. and Neoptolemos, J. P. (2012), The value of source data verification in a cancer clinical trial, *PLoS One* **7**(12), e51623.

Stute, W. (1995), The central limit theorem under random censorship, *The Annals of Statistics* 422–439.

Tsiatis, A. A. and Davidian, M. (2001), A semiparametric estimator for the proportional hazards model with longitudinal covariates measured with error, *Biometrika* **88**(2), 447–458.

Turchin, P. (1998), *Quantitative analysis of movement: measuring and modeling population redistribution in animals and plants*, Vol. 1, Sinauer Associates Sunderland.

Vantongelen, K., Rotmensz, N. and Van Der Schueren, E. (1989), Quality control of validity of data collected in clinical trials, *European Journal of Cancer* **25**(8), 1241–1247.

Weiskopf, N. G. and Weng, C. (2013), Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research, *Journal of the American Medical Informatics Association* **20**(1), 144–151.