Under which circumstances do simple methods for handling missing data in incomplete baseline covariates give reliable results: A simulation study and an application to EBMT CML study

Ge, J.

Under which circumstances do simple methods for handling missing data in incomplete baseline covariates give reliable results

a simulation study and an application to EBMT CML study

Junran Ge (S2497646)

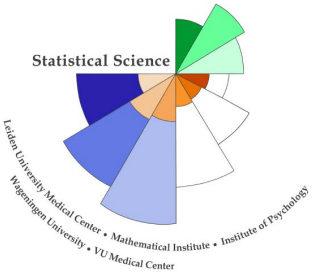
Thesis advisor: Prof.Hein Putter

External Supervisor: Luuk Gras (EBMT statistician)

MASTER THESIS

Defended on 27/10/2021

Specialization: Statistical and Data Science



Statistical Science for the Life and Behavioural Sciences

Abstract

Missing data is common in clinical research. How these missing values are handled has a direct impact on the final study results. Existing medical studies commonly use complete case analysis to remove observations with missing values, which has the advantage that it is simple and easy but depletes the information in the original dataset, and may result in biased estimated. Multiple imputation (MI) methods are often considered more reliable than complete case analysis, missing indicator methods and single imputation methods. However, recent research has shown that by comparing a number of MI methods, particularly where the underlying assumptions are undermined, some MI methods may cause more bias in model estimates than complete case analysis.

To study which methods would perform better under which circumstances, this thesis will perform a simulation study, comparing the results of the above-mentioned techniques under certain types of missingness, such as MCAR, MAR and MNAR. Complicated connections like missingness is also correlated with survival time will be considered.

The various parameter settings for the simulation study are based on a real case study where about 12% of the observations contain missing values for some variables. In addition to the basic MICE, two other multiple imputation methods are compared, one with interaction terms between the full variables and the baseline hazard in the imputation model, and the other with a specific substantive model in the iteration. In this thesis, the substantive model is Cox model. The simulation studies show that the one with interaction terms is not significantly different from MICE and its improvements are of limited applicability. The one with the specific substantive model is more suitable for complex data types and when there are strong correlations between covariates. Besides, basic MICE also performs well in data sets with a high proportion of missing binary covariates,

while the missing indicator method produces large bias in many settings, even for full case studies.

Keywords: missing values, overall survival, multiple imputation, alloHCT

Contents

1 Introduction

Missing values, common in epidemiological studies, are a major problem in obtaining valid estimates. Then there are some method used in solving missing values as following: complete cases analysis method which only uses the observations without any missing values, missing indicator method which need to add an variable indicates whether it is missing in each observation for the variable with missing and imputation methods. Simple imputation method replace the missing values by mean or median. Multiple imputation method which is based on Bayesian theory considers the imputed values are random and derived from the observed values. In general, multiple imputation methods are more effective than complete case analyses. However, under certain conditions, its bias may be greater than that of simpler methods such as complete case analysis. Groenwold [1] and his colleagues focused on the missing indicator method, comparing it with multiple imputation and complete case analysis, using real data with incomplete covariates from randomized and non-randomized studies. The results showed that in randomed trials the missing indicator approach worked better when the data were missing not at random (MNAR), while it resulted in biased estimates in non-randomized trials . In contrast, the complete data analysis method performed similar to the missing indicators methods when data were missing completely at random (MCAR). Similarly, Donders [2] showed that the missing indicator method would produce biased results when data are MCAR or missing at random (MAR), while complete data analysis would produce valid results when data are MCAR. Furthermore, multiple imputation was unbiased under both MCAR and MAR missingness mechanisms. White and Thompson [3] focused on the missing baseline data in randomized trials and suggest that missing indicator method was a good approach when the missingness of baseline hazard does predict the outcome.

However in 2009, White and Royston [4] analysed the plausibility of the imputation method from a theoretical perspective in conjunction with simulations. They imputated missing data in Cox models using new methods based on cumulative baselines or marginal hazards. Among the different multiple imputations,

those based on the Nelson-Aalen estimator had lower bias and higher power in most simulations. However, as covariates become more predictive of the results, all MI methods are likely to be biased. The reason is that imputation models are not entirely correct. Therefore, it can be concluded that MI is not always optimal in any situation. The fact could be considered as the true underlying models may be non-linear. The true underlying model is the analysis model whose regression coefficients are of substantive interest, also called substantive models. According to this, substantive model compatible fully conditional specification (SMC-FCS) is a new improving multiple imputation method. In original multiple imputation, a simple linear model is generally used to interpolate the missing values. Bartlett [5] proposed this new imputed way of incorporating the substantive model into the imputation model and pointed out that when the imputation model is specified exactly and the missing data mechanism is MAR, SMC-FCS will give the same results as MICE gives an estimate that is consistent with the results obtained from the complete data.

In a lot of EBMT studies, there are up to 70 % missing baseline data. The main aim of this thesis is to use a simulation study to compare the performance of multiple methods in different situations in order to discover when we can use simple methods (complete case studies or missing indicators) and when we need more complex methods (MI). For comparing which method is best for missing data in different situations of a simulation study, bias and root mean square error (RMSE) will be used to assess the merits of the various analyses. In Section 3, we will explain how we calculate these two evaluation indexes. Moreover, these methods will be studied in a simulation study and data from a real case study combined with findings of.

In this simulation study design, we use different scenrios in which we vary a number of factors. The first factor considered is missing data mechanism. In the missing data literature, three kinds of missingness patterns are considered (MCAR, MNAR and MAR, which are subdivided into those with missing variables that are only correlated with other complete covariates, and those that are correlated with both complete covariates and survival time). The second factor we vary are differ-

ences in strength of correlation between variables. It is important to note that we not only consider correlations between covariates, but also considered correlations between outcome (survival time and status) and covariates. The effects of other factors such as sample size and the proportion of patients with missing data are also studied.

In Chapter 2, we discuss the theory for several approaches of dealing with missing values. Chapter 3 describes simulation study, including the generation of missing data. Chapter 4 describes an application of these approaches to handling missing values to data from a real case EBMT study. Finally, we discuss the experimental results and state the limitations of this experiment in Chapter 4.

2 Method

2.1 Survival analysis

Survival analysis is a method of analysing survival time data and allows investigating the relationship between survival time and factors that are associated with it. It has a wide range of applications in many fields, such as the survival of a person or animal, a patient's condition being in remission (as opposed to relapse or deterioration), a system or product working properly (as opposed to failure or malfunction), or even the loss of customers in business. The endpoint event in this thesis is the time of the last occurrence before death or cessation of observation.

The two necessary components for survival analysis are whether the endpoint occurred (usually dichotomous, occurred or did not occur) and when the endpoint occurred (survival time) or,in case the endpoint has not occurred yet, the time the endpoint was last observed not to have occurred. This is when death has not been observed in a patient (possibly due to not long enough follow-up, or patients being lost to follow-up), and it is only known that the patient is still alive at a certain point in time, the survival time of the patient is said to be right-censored, which is the most common censored type and considered in this thesis.

More formally, the right censoring time C is known, not the actual but unknown survival time T. We assume that C is independent of T, and the final survival time is defined as $\hat{T} = \min(T,C)$. In non-censored data, the survival time T is the time elapsed from the start of the observation to the occurrence of the end event and is observable. To mark whether N observations are right-censored, status $D = I(T \leq C)$ is introduced, where $D = 0$ indicates a right-censored observation; each observation, $i \in \{1,2,...,N\}$ corresponds to $(\hat{t}_i, d_i)$.

There are two types of functions for survival analysis of central interest: hazard or risk functions and survival functions. The survival function is the probability, that the survival time of ab individual is greater to t, defined as $S(t) = P(T > t)$. When $t = 0$, the survival function has the value 1, and as time passes (the value of t increases), the value of the survival function becomes progressively smaller,

that is, a monotonically decreasing function of time t. Based on the above, the cumulative survival function is derived as $F(t) = 1 - S(t)$. The derivative of the cumulative survival function yields the risk probability function, which represents the probability of an event occurring at a point in time t:

$$f(t) = \frac{dF(t)}{dt} = \lim_{\Delta t \downarrow 0} \frac{F(t + \Delta t)}{\Delta t}$$

Ultimately, the hazard function can be obtained from the following calculation:

$$h(t) = \frac{f(t)}{S(t)} = \lim_{\Delta t \downarrow 0} \frac{S(t) - S(t + \Delta t)}{\Delta t \cdot S(t)}$$

It represents the probability of the event occurring at the next instant in time when T the patient has survived up to time point t.

A commonly used method for estimating $S(t)$ is the Kaplan-Meier method (KM curves). The idea is to write $S(t)$ as a recursive equation. We assume that we have calculated the value of the survival function $S(t_1)$ for time t_1 and want to calculate the value of the survival function for time t_2 $(t_2 > t_1)$ where t_2 is the next event time after t_1, then the individual first has to live past time t_1, expressed as follows:

$$S(t_2) = \text{prob(alive between } t_1 \text{ and } t_2) * S(t_1),$$
$$\text{with prob(alive between } t_1 \text{ and } t_2) = 1 - d/n,$$

where d represents the number of individuals that actually had an event at time t_2; n represents the total number of individuals that could have had an event at t_2 (which can be interpreted as the total number of individuals that are still alive and at risk just point to time t_2). Obviously, if no individuals had an event during the period from t_1 to t_2 $(d = 0)$, then the value of $S(t)$ would remain the same. So we only need to go through the recursive formula to update the value of $S(t)$ at time t when an event is observed to have occurred. The treatment of censored data need to be paid attention to. The number of objects at risk n, contains individuals censored at time t and after that in time, but no individual censored before that time point.

2.2 Cox proportional hazards model

Cox's proportional hazards (PH) model is one of the most widely used models used to study the relationship between various covariates and the hazard function and is a semi-parametric model. From the relationship between the survivor function and the hazard function an estimate of the survivor function can be found. It is then possible to make predictions of survival time for current or future patients with particular values for these factors. It only specifies the relationship between the influencing factors and the hazard function, and does not qualify the distribution of survival time. The risk function $h(t)$ can be expressed as follows:

$$h(t|X_i) = h_0(t)\exp(\beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p)$$

whereX_1,...,X_p are covariates and β_1,...,$\beta_p > 0$ are the coefficients to be estimated by the model. The greater the coefficient the shorter the survival time, while $\beta\_i < 0$ means protective factors. $exp(\beta_i)$ is called hazard ratio (HR). $h_0(t)$ is the baseline risk function, which is the risk function with all covariates at zero or the standard state. In Section 3.1, it will be discussed how to generate survival time by $h_0(t)$ with an assumed distribution.

The Cox model needs to satisfy the proportional hazard assumption, i.e. the HR is assumed not to change over time. Therefore, a PH hypothesis test is required after we fit the Cox model. Schoenfeld residuals can be used for validation [6].

2.3 Methods to deal with missing values

It is important to consider the reason for data being missing (e.g. new valuable variables are proposed with long studies. For these variables, the absence is clearly related to variables about time such as year.). Missing baseline values are usually classified into three categories according to the possible mechanisms of missing baseline values: missing completely at random (MCAR), missing at random (MAR) and missing not at random (MNAR). MCAR indicates that the probability of missing data occurring is independent of both observed and unobserved

data. If the probability of missing data occurring is related to the observed variable but not to the characteristics of the unobserved data, then it is MAR, while MNAR refers to non-negligible missing data if the missing data in the incomplete variable depend on both the complete variable and the incomplete variable itself. In realily, it's not possible to distinguish between MAR and MNAR, and judgements have to be made empirically.

In general, in statistical analyses missing values are usually handled by the deletion of patients with missing values or by imputing new values for the missing values. These two methods and other methods sometimes used are described below.

2.3.1 Complete case analysis

The deletion method is divided into removing samples or features where there are missing ones. This approach is also known as complete cases analysis (CCA), which is simple and easy to implement, but is recommended when the proportion of missing data is small, otherwise it can lead to bias and increase the bias, especially in the case of MNAR.

2.3.2 Missing indicator

As the name implies, this method creates a dependent variable to mark it as missing or not based on the variable that has the missing variable. If it is a continuous variable X, then an additional dependent categorical variable X' needs to be added, with 0 indicating normal and 1 indicating that it is a missing value. When X is a categorical variable, then a new category can be added to indicate a missing value, i.e. the missing value is defined as a new category. In the remainder of this thesis, we abbreviate missing indicator as MID.

2.3.3 Multiple imputation

The idea behind imputation comes from the idea that imputing missing values with the most likely values produces less information loss than removing patients with

partially missing data altogether. Typical examples are single imputation, such as mean or median imputation that use mean or median of the observed values. This method is simple and easy to implement. However, by pretending you have observed values which were missing, the standard errors of estimate are most likely to be too small. In an attempt to solve the problem of standard errors being too small, Rubin [7] proposed in the 1970s not to impute a single value but multiple values.

If we impute m times a value we end up with, m different dataset which are all separately analysed. All m estimates are then combined into 1 estimate and a standard error according to Rubin's rule. This is usually done in three steps as follows in Figure 1:

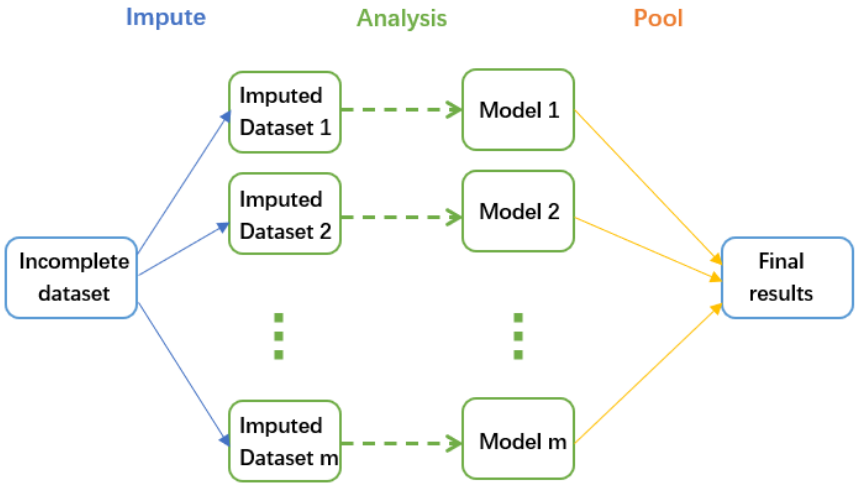


Figure 1: Progress of multiple imputation

Imputation: For each missing value, it produces a set of possible imputation values that reflects the uncertainty (noisy); each value can be used to impute missing values in the data set, producing m complete data sets. *Analysis*: Each imputed data set is statistically analysed using statistical methods specific to the complete data set. As a result, m sets of parameter estimates $\hat{\beta}_i$ are obtained along with the corresponding standard error. *Pool*: The results from each imputed data set are selected according to Rubin's Rules[7] to produce the final imputed values. For

example, the pooled parameter estimate for the parameter β_i is

$$\bar{\beta}_i = \frac{1}{m}\sum_{j=1}^{m}\hat{\beta}_{ij}$$

The variance of parameter estimators is divided into within imputation variance (represented by the confidence intervals):

$$\bar{U}_i = \frac{1}{m}\sum_{j=1}^{m}\hat{U}_{ij}$$

and between imputation variance (horizontal shift between imputations):

$$B_i = \frac{1}{m-1}\sum_{j=1}^{m}(\hat{\beta}_{ij}-\bar{\beta}_i)^T(\hat{\beta}_{ij}-\bar{\beta}_i)$$

The overall variance and $(1-\alpha)100\%$ confidence interval (CI) referenced a t-distribution of pooled parameter estimators are

$$V_i = \bar{U}_i + B_i + B_i/m$$

$$\bar{\beta}_i \pm t_{\nu_i}(\alpha/2)\sqrt{V_i}, \quad \text{degrees of freedom} \nu_i = (m-1)(1+\frac{\bar{U}_i}{B_i+B_i/m})^2$$

In the imputation section, if we have only one variable with missing values, i.e. univariate missing data, if we apply the fitted function directly to imputation, imputed values do not take into account the added uncertainty and for each missing value. As an example, we assume there are three variables X_1, X_2, X_3, of which only X_2 contains missing values. First, the complete data is fitted: $x_{i2} = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i3} + \varepsilon_i$,$x_2 isacontinuousvariable$, then each missing value is imputed based on the value of the other variable in the observation: $\hat{x}_{i2} = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i3}$. This method can be further improved by taking into account the standard error of the parameter estimators and the uncertainty of the imputed value (variation of the residuals). The final value can be taken as its mean and the variation can be obtained directly.

However, when missing values occur in more than one variable, the imputation process is a challenge. In conjunction with the Markov chain Monte Carlo (MCMC) idea, there are two ways of imputing multivariate data: joint modelling (JM) and fully conditional specification (FCS), also known as multivariate imputation by chained equations (MICE). JM, developed by Schafer [8], requires specifying a multivariate distribution containing missing values and using MCMC to imputed from the conditional distribution. In contrast, MICE imputes multivariate missing data on a variable-by-variable basis, based on the univariate missing data imputation described above. in the absence of a suitable multivariate distribution, MICE clearly outperforms JM. This paper selects MICE and one of its modifications, substantive model compatible fully conditional specification (SMC-FCS). The details of this two methods are as follows.

2.3.3.1 MICE

MICE assumes that the missing data is MAR[9], which means the probability that a value is missing is dependent on other observed values nd not on unobserved values, so it is also possible to predict this missing value from other values. We consider $X = \{X_1, X_2, ..., X_p\}$ as the p partially observed covariates and let $X_{-j} = \{X_1, X_2, ...X_{j-1}, X_{j+1}, ..., X_p\}$, q fully observed covariates $Z = \{Z_1, Z_2, ..., Z_q\}$ and fully observed outcome Y. We assumed a missing indicator matrix R, where 0 means that the corresponding position in X is missing and 1 means that there is an observed value. Observed and missing in X are denoted by X^{obs} and X^{mis} respectively. Then the MAR hypothesis can be expressed as: $P(R|Y,X,Z) = P(R|Y,X^{obs},Z)$. The parameters are denoted as θ.

Combined with the above process of doing imputation for univariate missing data, the MICE process is as follows:

MICE algorithm for *h*th imputed dataset

1.Starting imputations X_j^0 is randomly drawn from X_j^{obs}

2.The posterior distribution of θ is sampled iteratively from conditional distributions of the form:

$P(X_i|X_{-i},Z,Y,\theta_i), i \in 1,...,p$

3.**for** t in 1,...,q:

for j in 1,...,p;

Draw parameters $\theta_j^{*t} \sim P(\theta_j^t|X_j^{obs},Z,Y,X_{-j}^{*(t-1)},R)$

Draw imputations $X_j^{*t} \sim P(X_j^{mis}|X_{-j}^{*(t-1)},Z,Y,R,\theta_j^{*t})$

end for

end for

Here, q need to satisfy convergence, i.e. the sampling distribution does not change any more. But as the previous imputation of X_j^{*t} goes to the next loop iteration by association with other variables, rather than directly. This speeds up convergence, so the value of q can be a very small value, and in Chapter 3 we choose $q = 2$. Although it seems to be quite small, it works fine most time in Chapter 3. Once the above process is complete, an imputed data set is formed. To obtain m imputed data sets, the above process needs to be run m times.

Under the Cox model, we use two imputation models to impute missing Xs. Firstly, with the log-likelihood of outcomes T, D and Bayes' algorithm, we could get the conditional distribution of X with complete covariates Z:

$$\begin{aligned} logp(\hat{T},D|X,Z) &= Dlogh(\hat{T}|X,Z) - H(\hat{T}|X,Z) \\ &= D(logh_0(\hat{T}) + \beta_X X + \beta_Z Z) - H_0(T)exp(\beta_X X + \beta_Z Z), \\ logp(X|\hat{T},D,Z) &= logp(X|Z) + D(\beta_X X + \beta_Z Z) - H_0(\hat{T})exp(\beta_X X + \beta_Z Z) + const \\ &\doteq \alpha_0 + \alpha_1 D + \alpha_2 H_0(\hat{T}) + \alpha_3 Z. \end{aligned}$$

Here, the Cox's PH model is $h(t|X,Z) = h_0(t)exp(\beta_X X + \beta_Z Z)$ and α is equal to θ in the MICE algorithm above. For increasing the accuracy of approximation, an interaction term $\alpha_4 Z * H_0(\hat{T})$ could be added in the formula.

2.3.3.2 SMC-FCS

One of the advantages of MI is that it separates the process of dealing with the missing values (the Imputation part) from the analysis of the completed data (the Analysis part). However this division could cause bias. In the case of imputation of partially observed covariates, the imputation may be generated from a model that is incompatible with the substantive model, which may lead to biased parameter estimates (asymptotic approximations) in the analysis part. According to the definition of compatibility given by Liu et al.[10], when there is no condition equal to the joint model of two conditional models, such two conditional models are incompatible. If the substantive model contains non-linear covariates or interactions of covariates, the imputation model chosen by default will be incompatible with the substantive model. For example, we usually use the logit model as an imputation model to impute the missing values in binary variables. However, if the substantive model (true underlying model) contains an interaction term of this binary variable, i.e. this binary variable interacts with other variables and the interaction has an effect on the outcome (survival time and status), the default logit model does not take this interaction into account and could impute biased data. In order to avoid incompatibility between univariate imputation models and solid models in MICE, Bartlett et al.[11] proposed substantive model compatible fully conditional specification (SMC-FCS).

Based on Bayes' theorem, the conditional distribution can be expressed as:

$$\begin{aligned} P(X_j|X_{-j},Z,Y) &= \frac{P(Y,X_j|X_{-j},Z)}{P(Y|X_{-j},Z)} \\ &= \frac{P(Y|X_j,X_{-j},Z)P(X_j|X_{-j},Z)}{P(Y|X_{-j},Z)} \\ &\propto P(Y|X,Z)P(X_j|X_{-j},Z) \end{aligned}$$

Thus, using the density proportion $P(Y|X,Z,\psi)P(X_j|X_{-j},Z,\theta_j)$ to impute X_j could make the imputation model automatically be compatible with the substantive model $P(Y|X,Z,\psi)$, where ψ are the parameters of substantive model and θ_j is a vector of imputation model parameters for given jth iteration. The choice of model

$P(X_j|X_{-j},Z)$ could be same with the normal FCS above. At the tth iteration, SMC-FCS algorithm imputes missing values in X_j depending on both θ_j and the substantive model parameters ψ, by performing the following draws:

$$\theta_j^t \sim f(\theta_j)P(X_j^{mis}, X_j obs|X_{-j}^{*t}, Z, \theta_j)$$
$$\psi^{(t,j)} \sim f(\psi)P(Y|X_j^{mis}, X_j^{obs}, X_{-j}^{*t}, Z, \psi)$$

where $f(\theta_j)$ and $f(\psi)$ denote uninformative priors. Being similar with MICE, the SMC-FCS algorithm uses a random selection from the observed values as the initial value. In this paper, the substantive model is a Cox model, then outcome Y is survival time $\hat{T} = \min(T,C)$ as well as status D, and the substantive model can be expressed as $h(t|X) = h_0(t)f(X_j, X_{-j}, Z, \beta)$. The baseline hazard $h_0(t)$ is represented parametrically by a finite set of parameters λ, so that $\psi = (\beta, \lambda)$. In summary, SMC-FCS adds elements of the substantive model to the imputation model of simple MICE, so that the data generated is more closely aligned with the substantive model that we subsequently use to analyse.

3 Simulation

In each Monte Carlo replication, we generate N (sample size) individuals. The whole simulation process can be summarized as follows:

Step 0: For each N_{rep} Monte Carlo replications:

Step 1: Generate Covariates Z_c, Z_b, X_c, X_b.

Z means it is a complete variable and X means it has missing values. c means continuous variables and b means binary variables. Pay attention to the correlation between these covariates (Section 3.1.1)

Step 2: Generate Survival time and status.

Use weibull distribution (Section 3.1.2) depending on covariates sampled in the previous step. Censoring time is generated from exponential distribution.

Step 3: Model fitting I.

Calculate the parameters and CI based on the complete data, set as benchmark.

Step 4: Generate missing values in X_c, X_b.

The main scenarios are MCAR, MNAR and MAR.(Section 3.13)

Step 5: Model fitting II.

Use the five methods presented in Section 2 in the amputed dataset, and parameters and CIs are calculated based on the processed dataset.

Step 6: Evaluation.

Compute bias and RMSE by using the results of Step 3 and Step 5.

bias of coefficient β_i: $\frac{1}{N_{rep}} \sum_j^{N_{rep}} (\hat{\beta}_{ij} - \beta_i)$

RMSE of coefficient β_i: $\sqrt{\frac{\sum_j^{N_{rep}} (\hat{\beta}_{ij} - \beta_i)^2}{N_{rep}}}$

The bias of the parameter estimates are given prior to compare. The definition

of the Monte Carlo standard error for the bias is:

$$\text{MCSE} = \sqrt{\frac{1}{N_{rep}(N_{rep}-1)}\sum_{i=1}^{N_{rep}}(\hat{\beta}_i - \beta)^2} \doteq \sqrt{\frac{\text{variance of } \hat{\beta}}{N_{rep}}}.$$

The number of repetitions N_{rep} is set to 100, taking into account the time of the experiment. We found that the standard errors $SE(\hat{\beta}_1)$, $SE(\hat{\beta}_2)$ of the estimated $\hat{\beta}_1$ and $\hat{\beta}_2$ are 0.149 and 0.107 respectively in the base scenario with complete data. These are the largest observed empirical standard errors in small trials. According to these, $\text{MCSE}(\beta_1) = 0.0149$ and $\text{MCSE}(\beta_1) = 0.0107$.

This simulation study was conducted using version R3.6.3 [12]. Survival models as well as proportional hazard models were analysed using the coxed package version 0.3.3 [13] as well as the survival package version 3.1.12 [14]. Incomplete data were imputed based on the proportional hazard model using the smcfcs package version 1.5.0 [11] and missing data were generated and imputed using the mice package version 3.9.0 [15]. Modifications applicable to this simulation study were made based on the PoisBinOrdNor package version 1.6.3 [16], resulting in the transformation of the correlation coefficients.

3.1 Data-generating mechanisms

3.1.1 Covariates

In each Monte Carlo replication in this simulation, there are four variables, two are complete, labelled Z, and the other two are generated and replaced partially in Section 3.1.3 for missing values, labelled X. Of the two complete Zs, one is the continuous variable Z_c and the other is the binary variable Z_b. Similarly, the two Xs are X_c, X_b.

These parameters are estimated based on real case data. To represent the age of patients and donor, $Z_c \sim N(45.87, 12.56)$ and $X_c \sim N(36.74, 12.50)$. Referring to the Karnofsky score, the CMV status categorical variable, $X_b \sim Bern(0.25)$ and $Z_b \sim Bern(0.33)$. Before fitting substantive model for analysis, two continuous

covariates X_c and Z_c are scaled as follows: $(X_c - 45)/10$ and $(Z_c - 45)/10$.

Given that covariates are potentially correlated with each other, two correlated continuous-type variables can be sampled from a binary normal distribution. However, it is somewhat tricky to take into account the correlation between binary and continuous variables in the generation. Referring to the point-biserial correlation proposed by Demirtas et al[17]. it is possible to generate covariates that are eligible. In this paper the correlation between covariates was all set to ρ and varied to $\rho = \{0.01, 0.2, 0.5\}$ in the simulations.

In brief, four variables are generated using a multivariate normal distribution based on a specific correlation matrix. Two of these variables are transformed into binary variables using the dichotomous method. Then we need to convert some correlation coefficients ρ into new correlation coefficients. We assume X_1, X_2 both follow standard normal distribution $N(0,1)$, and $X_3 = I(X_1 > k) \sim Bern(p)$. Here, k is dichotomization threshold, and when $X_1 \sim N(0,1)$, $P(X_1 < k) = p$. Thus, $E(X_3) = p, Var(X_3) = p(1-p)$. If δ_{12} denotes the correlation between X_1 and X_2, then $X_2 = \delta_{12} X_1 + \varepsilon$, $\varepsilon \sim N(0, 1-\delta_{12}^2)$. Further, the correlation between X_3 and X_1 δ_{13}, and the correlation between X_3 and X_2 δ_{23} can be calculated as follows:

$$\begin{aligned}
\delta_{13} = \frac{Cov(X_1, X_3)}{\sqrt{Var(X_1)Var(X_3)}} &= E(X_1, X_3)/\sqrt{p(1-p)} = f_{N(0,1)}(k)/\sqrt{p(1-p)}, \\
\rho = \delta_{23} = Cov(X_2, X_3) &= Cov(\delta_{12} X_1 + \varepsilon, X_3) \\
&= \delta_{12} Cov(X_1, X_3) + Cov(X_3, \varepsilon) = \delta_{12}\delta_{13},
\end{aligned}$$

$f_{N(0,1)}(k)$ denotes the value of standard normal distribution density function $f(x)$ when $x = k$.

Moreover, if X_1 and X_2 are both binary variables, phi correlation Φ (Pearson correlation applied to dichotomous data) is used. It is calculated as follows:

Step 1: Calculate a two by two contingency table.

	$X_2 = 0$	$X_2 = 1$
$X_1 = 0$	A	B
$X_1 = 1$	C	D

Step 2: $\Phi = (AD - BC)/\sqrt{(A+B)(C+D)(A+C)(B+D)} = \rho$

After that, the "phi2tetra" function in R can be used to convert the phi correlation into the desired correlation. By using the method described above, the original correlation coefficient matrix can be transformed into a correlation coefficient matrix for generating multivariate normal distributions.

3.1.2 Survival time and status

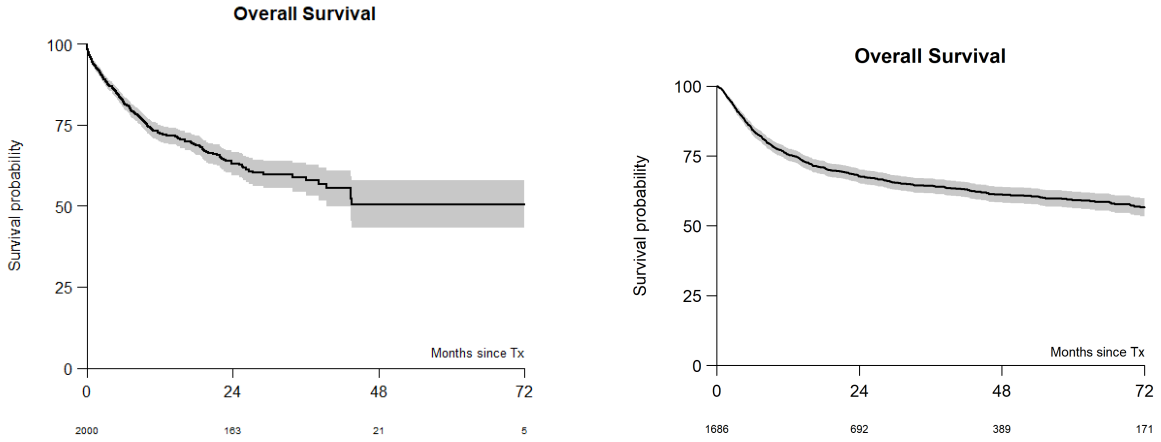
By observing the real cases in Chapter 4, the assumption of a constant hazard function does not hold for the data set in alloHCT studies. And in the initial part, the risk rises to a high value, then decreases when the survial time is longer. Therefore, the Weibull distribution, which is characterised by two positive parameters, is used to sample the lifetime T. In $Weibull(\lambda, \alpha)$, the parameter λ is known as the rate parameter, while α is the shape parameter. Hazard function which is introduced above can be expressed as $h(t|X) = h_0(t)exp(\beta_1(Z_c + X_c) + \beta_2(Z_b + X_b))$, Cumulative hazard funtion is $H_0(t) = \lambda t^{\alpha}$. According to the probability density function of the Weibull distribution, the baseline hazard can be written as $h_0(t) = \lambda \alpha t^{(\alpha-1)}$. For $\alpha > 1$, the hazard function increases and for $0 < \alpha < 1$, the hazard function decreases.

Let Y be a random variable following the distribution function F, then we have $U = F(Y)$ following a uniform distribution with interval $[0, 1]$. Combining this with $1 - U \sim Unif(0, 1)$, the generation algorithm of T could be derived as fol-

lows:

$$
\begin{aligned}
1-U &= exp(-H_0(t)exp(\beta_1(Z_c+X_c)+\beta_2(X_b+Z_b))),\\
T &= H_0^{-1}(\frac{-log(1-U)}{exp(\beta_1(Z_c+X_c)+\beta_2(X_b+Z_b))})\\
&= (\frac{-log(1-U)}{\lambda\cdot exp(\beta_1(Z_c+X_c)+\beta_2(X_b+Z_b))})^{1/\alpha}.
\end{aligned}
$$

At the same time, censoring time C is independently generated from a exponential distribution with parameter λ_C, i.e. $C \sim Exp(\lambda_C)$. The survival time is denoted as $\hat{T} = min(T,C)$ with status $D = I(T \leq C)$. In this simulation study, we set $\lambda = 0.044, \alpha = 0.634, \lambda_C = 0.08$ which are estimated by Maximum Likelihood function with real case data. The formulas of this part could be found in Appendix. Regarding the coefficients, we varied $\beta_1 = \{0.05, 0.2, 0.5\}$ and fixed $\beta_2 = 0.5$ in the simulations.



(a) KM curves of a sampling dataset (b) KM curves of real case dataset

Figure 2: KM curves

Figure 2 shows the KM curves of survival time in one simulated dataset and the real dataset separately. It could be seen that the trends are same, but the number of individuals in tails is quite small. Only 5 individuals exist at the time point 72 months. If we check the density of survival times in the real dataset in Figure 3, it is obviously different from a normal Weibull distribution (blue line). In the right tail, it changes to have more individuals instead of being infinitely close to 0.

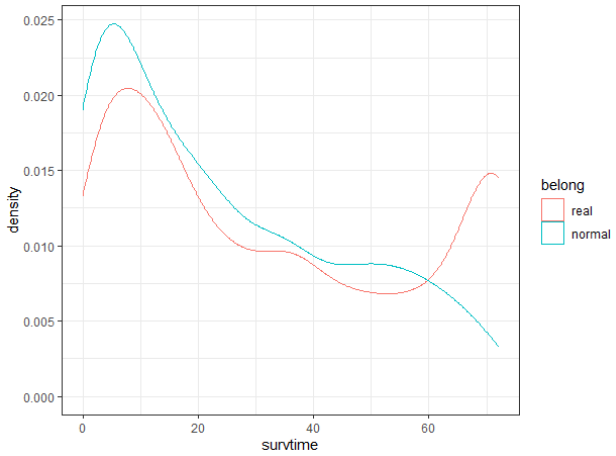


Figure 3: Density plot of survival time in real dataset.

3.1.3 Missing data mechanisms

In the simulation experiments in this paper, there are two variables X_c and X_b that need to be replaced with some of their missing values before proceeding to the next step in the analysis. Simple univariate amputation procedures typically do amputation on one variable at a time, i.e. the missing values are now generated in X_c, followed by X_b. Although there are a number of modified univariate amputation procedures, all are specific to a particular data set.

Consider that the causes of missingness are not only independently random, but may be correlated with each other, depend on values of covariates, or depend on the variables themselves. The proportion of missingness is hardly guaranteed to be the same under different missingness mechanisms because of the undergone of missingness mechanism. Therefore, this thesis uses the multivariate amputation method proposed by Rianne Schouten [18], and the disadvantages of using more univariate amputation procedures she also elaborates on. Figure 4 shows how to ampute a dataset with 4 variables and N individuals.

First, we need a pattern matrix to illustrate the possible forms of missing values, namely X_c only is missing, X_b only is missing, or both X_c and X_b are missing. Here, we set the probability of occurrence to be equal for all three cases. If we use 0 to

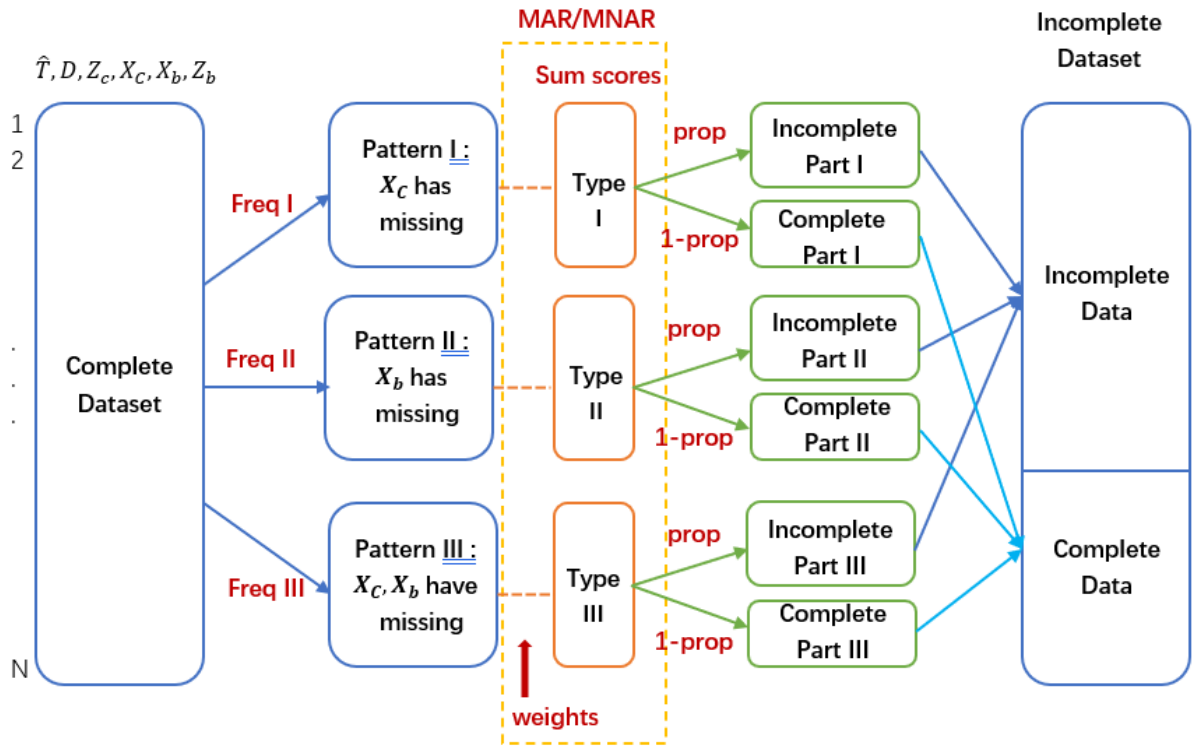


Figure 4: Progress of amputation. In this simulation study, Freq I = Freq II = Freq III = $1/3$. Thus, the subsets of three patterns have same size respectively.

mark non-missing and 1 for missing, we get the following pattern matrix:

$$T_{pat} = \begin{array}{c} \begin{array}{cccc} Z_c & X_c & X_b & Z_b \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Depending on the number of patterns, the original dataset will first be divided into 3 random subsets. The size of each subset is then determined by the likelihood of each pattern occurring. Since we set them equal, the three subsets in this paper are approximately equal in size.

The concept of weighted sum scores was introduced in order to link missing values with other value relationships. According to the weighted sum scores, each observation is given a probability of being missing. Weighted sum scores are calculated as the outcome of a linear regression equation of the observed values

$\{Z_c, X_c, X_b, Z_b\}$ according to pattern matrix T_{pat}, where the coefficients are determined by the researcher. The weighted sum score of observation i is calculated as follows:

$$wss_i = \omega_1 Z_{ci} + \omega_2 X_{ci} + \omega_3 X_{bi} + \omega_4 Z_{bi},$$

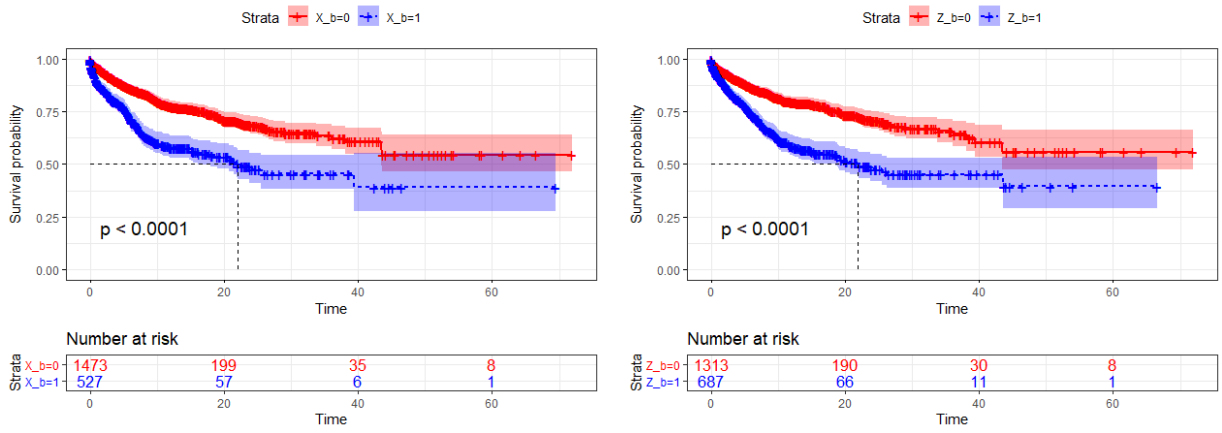
where $\{\omega_1, \omega_2, \omega_3, \omega_4\}$ are the corresponding pre-specified weights. The weight of each variable controls its effect on the weighted sum score. The sign of the weight value is also meaningful. Positive weights make the weighted sum score larger and thus increase the probability of being missing, while negative weights have the opposite effect. If the absolute value of the variable weight is large, this means that the variable has a greater effect on missingness. Therefore, variables that are more strongly correlated with missing variables have greater weights. By adjusting for differences in weights, the type of missingness can then be adjusted. When all ω equal to zero, no variable would play a role in generating missing values. In the case of MCAR, all weights are equal to 0. If we want to generate data as MAR, we can do this by setting the weights of the missing variables to zero and only adjusting the weights of the complete variables. When only the missing variable has a non-zero weight, the data is MNAR. If the missingness is connected with survival time, a term $\omega_5 T$ could be added in the formula.

In this thesis, we set all the weights uniformly to ω. When we want to generate MAR data, the weight of the complete variable is ω. And when the data is MNAR, the weight of the missing variable is ω. In the MCAR condition, there will be no different scenarios in terms of weights which are all equal to 0. In addition, in the MAR scenario, we can also take survival time into account and divide it into those where only the full variable has an effect and those where both the full variable and survival time have an effect. Because the missing proportion of MAR missingness depends on the value of the observed variable, we can generate a MAR missingness data by assigning 0 weights to all variables that will be amputed. In contrast, if we choose to assign a non-zero weight to one or more of the variables that will be amputed, the missingness mechanism generated becomes MNAR.

3.2 Scenarios

We compared the performance of the different methods in a variety of setting data, Figure 5 shows the factors which were varied in the different scenarios. Each factor has a base value. This value is based on values observed in the real case study. When one factor from the baseline model is changed, all other factors remain the same with these base values. This means that we only change one factor at a time. In imputation part, continuous covariates were imputed using linear regression and binary covariates were imputed using logistic regression.

Figure 5: Different scenarios parameter setting.Each point would be amputed missing values under MCAR, MNAR, MAR and MAR-T missingness mechanisms and would be analyzed with 5 methods.



(a) KM curves of X_b (b) KM curves of Z_b

Figure 6: KM curves of a simulated dataset

3.3 Results

Before proceeding with the overall model building, we first consider whether the generated variables have different risks at different levels. Therefore, we performed a simple univariate analysis using KM curves. In Figure 6, it can be observed from the KM curves constructed for the two categorical variables that different categories of the same variable do have different survival functions. The p-value in the graph is the result of the log-rank rank test, where less than 0.05 means that the difference between the different groups is statistically significant.

Table 1 and 2 shows the bias and RMSE of the results which are obtained in different scenarios when using different methods under MCAR missingness. Table 3 and 4 shows the bias and RMSE of the results which are obtained in different scenarios when using different methods under MAR-T missingness. Tables of bias and RMSE values for MAR, and MNAR are included in the Appendix A. By comparing bias and RMSE, the inclusion of interaction terms in the imputation model of MICE does not reduce it significantly, namely column MI and MI-Int are quite same, so the subsequent discussion combines MI and MI-Int into a unified discussion of MICE.

CCA performs well in most scenarios of MCAR, especially in estimating the parameters of dichotomous variables, like column CCA compared with other columns

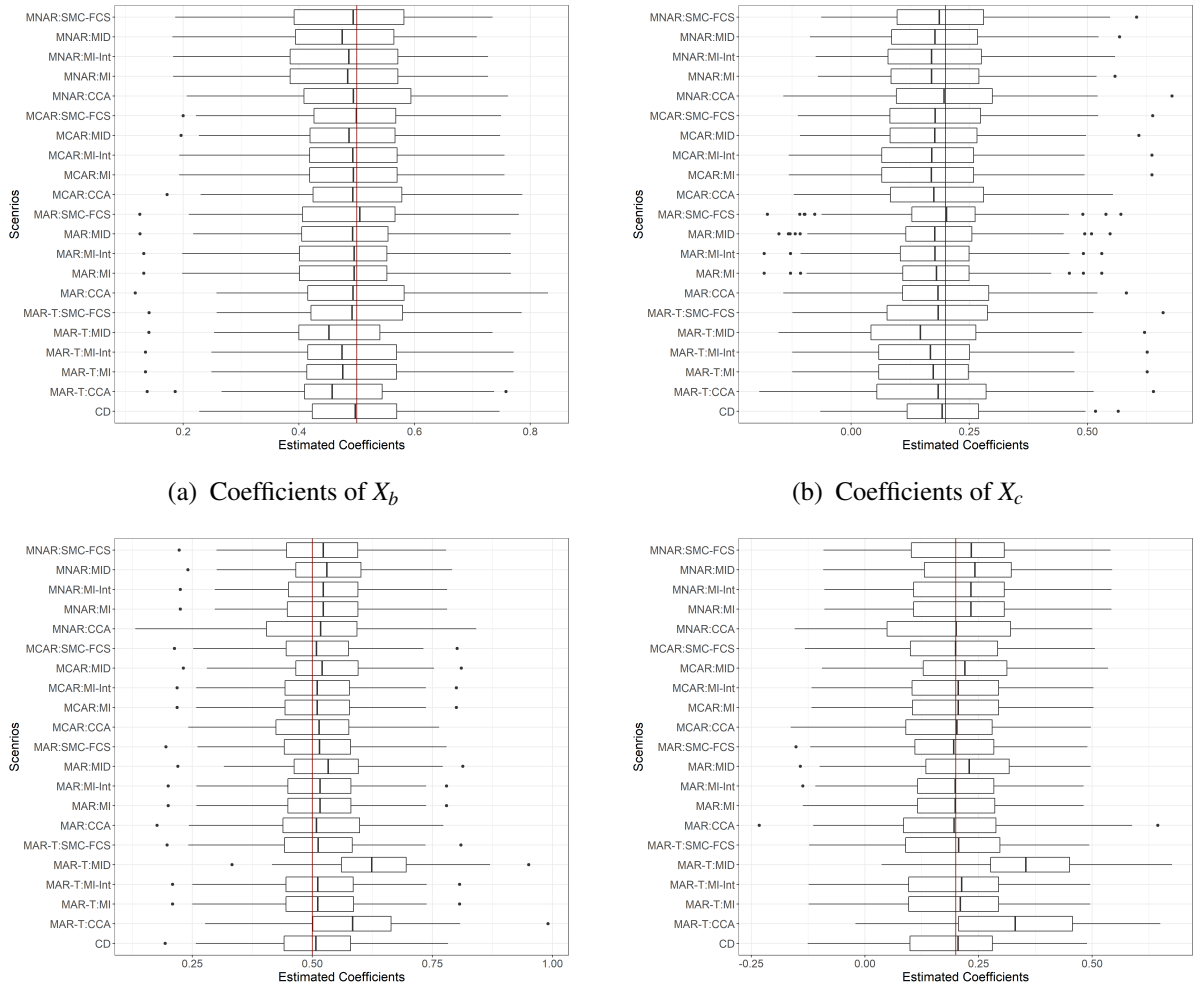
in X_b part in Table 1 are smaller except under small sample size dateset. MICE and MID are less biased when the data set is small. In particular, MICE gives better parameter estimates for the missing variables, while MID gives better parameter estimators for the complete variables. At the same time, there is a significant increase in the bias of analysis with complete data (CD) with increasing percentage of mssingness indicating that the overall bias of parameter estimates increases significantly when sample size decreases due to an increase in MCSE, which corroborates Morris and White's findings [19]. The bias in MID is severe on all covariates when the proportion of missing observations is elevated, and while bias in SMC-FCS is severe only on variables containing missing values. Although the bias estimated by MICE was smaller than that of CCA on the complete variables, it was much larger in the missing variables, especially when there are missing dichotomous variables, than in CCA (about 80%).

Correlation between covariates can also have an impact on accuracy. When the correlations between covariates are strong, SMC-FCS produces the least bias, followed by CCA, both are almost unbiased. Meanwhile, MID showed a significant increase in bias with increasing correlation. There is a similar increasing trend for MICE, but its utility is comparable to SMC-FCS for dichotomous variables with missing covariates. In terms of the values of the parameters, the bias of the SMC-FCS is smaller when the actual parameter estimators are larger, i.e. when the HRs are larger. The bias of MICE does not change very much when the HR increases, except for continuous variables with missing variables. This could be seen in the rows of strength of association in Table 1.

MCAR can be seen as a special missingness mechanism of MAR. If we compared Table 1 and 2 with the tables of MAR in Appendix A, the almost unbiased case mentioned above can only be called less biased at this point due to the relationship with other covariates, despite the same trend in bias. And when the actual parameter values change, the bias produced by MICE also changes significantly, although it is still the SMC-FCS that performs most consistently and optimally.

When in the presence of MAR-T missingness (the results are in Table 3 and 4), the bias of CCA is noticeable in many scenarios. First, for the complete continuous

variable Z_c, CCA and MID produced significantly biased results in eight different scenarios. In addition, for the parameter estimators of the other complete variables, the MID results are also strongly biased. SMC-FCS is better than MICE only when the correlations are strong (between covariates and between continuous covariates Z_c, X_c and outcome). Other times MICE produces relatively small biases, but the differences are quite small that the differences of bias are around 0.01. The results for the other covariates show that CCA is likely to outperform SMC-FCS when the sample size is very small. However, for larger sample sizes, the least biased method changes from CCA to SMC-FCS.



(a) Coefficients of X_b

(b) Coefficients of X_c

(c) Coefficients of Z_b

(d) Coefficients of Z_c

Figure 7: Box plots of Monte Carlo estimated coefficients in different base scenarios. Red line means the true value. The left and right line of the boxes mean the 25th and 75th percentile. The points are outliers and the bold short black lines are medians. X_b and X_c are the binary and continuous variable with missing values. Z_b and Z_c are the complete binary and continuous variable.

When missingness is MNAR, when the factor is related to increase correlation, the bias increases more significantly than the MAR. When the correlation between covariates increased, CCA produced less bias than SMC-FCS, especially the parameters of categorical variables.

When comparing RMSEs, the differences between the several methods are small and the RMSE for CCA is usually slightly larger. The value of the RMSE incorporates variance and bias. Therefore, although CCA produces small bias at times, its variance is large and the actual results are likely to be in its confidence interval. In the actual case analysis, attention needs to be paid to the confidence intervals of the parameter estimates.

The results of the above bias can be due to individual specific replications.Therefore all 100 individual simulation estimates from the baseline scenario are shown in the box plot in Figure 6. It is clear that the MID and CCA bias is relatively large, especially with MAR-T missingness. In (c) and (d) of Figure 7, the boxes (interquartile) of MID and CCA in Z_b and Z_c under MAR-T missingness are totally deviate from the true value. SMC-FCS performs a little better than MICE. However, when it comes to MNAR missingness, CCA is the method with less bias even than SMC-FCS. In both the MAR and MCAR missingness, SMC-FCS generally produces less biased results, while MICE and CCA are slightly more biased, but still within acceptable limits.

Other box plots are shown in the Appendix B. In the case of smaller sample sizes, CCA produces significantly larger deviations. At the same time, MID performs slightly better in some cases with missing variables, but there is still a significant bias in the parameter estimates for the complete variables. In addition, the performance of MICE and SMC-FCS is comparable for smaller data sizes. When the correlation between the variables themselves is small, it can be found that the overall parameter estimates are closer to the true values. When the correlation between the variables themselves is higher, SMC-FCS and MICE are the methods that produce less bias. Furthermore, it could be concludes that CCA and MID produce larger biases in MAR-T when it is estimating the parameters of the complete variables.

Table 1: Bias of the estimated coefficients (comparing with $\beta = \beta_1, \beta_2 = 0.5$) for MCAR.

		Binary missing (X_b)						Continuous missing (X_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		-0.0036	-0.0013	-0.015	-0.0107	-0.0108	-0.0076	-0.0033	-0.0104	-0.0199	-0.0251	-0.0247	-0.0107
sample	200	0.0182	0.0159	0.0032	0.0014	0.0014	0.006	0.0356	0.0263	0.0152	0.0018	0.0018	0.0137
size	500	0.0306	0.028	0.0204	0.0218	0.0218	0.0276	-0.0268	-0.0298	-0.0438	-0.0479	-0.0479	-0.0321
corr between	0.01	-0.0078	-0.0039	-0.0066	-0.0077	-0.0078	-0.0081	-0.0088	-0.0107	-0.0128	-0.0161	-0.0163	-0.0151
covariates	0.5	0.0029	0.0006	-0.033	0.0027	0.0029	-0.003	0.0028	-0.0007	-0.0026	0.0328	0.0329	-0.0046
strength of	0.05	-0.0046	-0.0026	-0.0125	-0.0082	-0.0081	-0.0073	-0.0031	-0.0059	-0.0134	-0.0204	-0.0202	-0.0057
associations	0.5	0.0046	0.0058	-0.0104	-0.0088	-0.0088	-0.001	-0.001	-0.005	-0.0172	-0.0249	-0.0249	-0.0064
β_1													
missing	60%	-0.0036	0.0001	-0.0373	-0.0244	-0.025	-0.0469	-0.0033	0.0077	-0.0478	-0.0561	-0.057	-0.0425
proportion													
		Binary complete (Z_b)						Continuous complete (Z_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		0.0062	0.0083	0.0264	0.0105	0.0104	0.0083	-0.0074	-0.006	0.0203	-0.0016	-0.0017	-0.0045
sample	200	-0.0335	-0.0084	-0.0079	-0.0238	-0.0238	-0.0253	-0.0182	-0.0423	0.002	-0.0057	-0.0057	-0.011
size	500	0.0043	-0.0147	0.023	0.0069	0.0069	0.0048	-0.0193	0.0063	0.0098	-0.0143	-0.0143	-0.0181
corr between	0.01	0.0076	0.0094	0.0079	0.0066	0.0066	0.0072	-0.0058	-0.0061	-0.0058	-0.0066	-0.0065	-0.0067
covariates (ρ)	0.5	-0.0109	-0.009	0.0272	-0.0176	-0.0178	-0.0051	-0.0055	-0.0029	0.0517	-0.0164	-0.0164	-0.001
strength of	0.05	0.006	0.0084	0.022	0.0093	0.0092	0.0075	-0.0083	-0.0046	0.0138	-0.004	-0.0042	-0.007
associations	0.5	-0.0005	0.0021	0.0263	0.0047	0.0047	0.0018	-0.0155	-0.0128	0.0207	-0.0084	-0.0085	-0.0126
β_1													
missing	60%	0.0062	-0.0224	0.062	0.0153	0.0154	0.0269	-0.0074	-0.0123	0.0732	0.0086	0.0092	0.024
proportion													
(prop)													

CD: estimators in complete data, CCA: complete cases analysis, MID: missing indicator, MI: MICE,
MI-Int: MICE with interaction of complete variables Z and hazard $H_0(\hat{T})$
When each factor is varied, the results of the baseline need to be added for comparison.
The number of replications is 100. In baseline model, sample size is 2000; correlations between the covariates are equal to 0.2; $\beta_1 = 0.2$;missing proportion is 20%

Table 2: RMSE of the estimated coefficients (comparing with $\beta = \beta_1, \beta_2 = 0.5$) for MCAR.

		Binary missing (X_b)						Continuous missing (X_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		0.1091	0.1258	0.125	0.1242	0.124	0.1243	0.129	0.1426	0.1395	0.1388	0.1389	0.1394
sample	200	0.356	0.4005	0.3718	0.3717	0.3717	0.3703	0.4751	0.4982	0.5057	0.4794	0.4794	0.4934
size	500	0.2238	0.2304	0.2203	0.2249	0.2249	0.2248	0.2646	0.3091	0.2957	0.2996	0.2996	0.2973
corr between	0.01	0.104	0.1167	0.1141	0.1149	0.1149	0.114	0.1213	0.1344	0.1312	0.1302	0.1304	0.1317
covariates	0.5	0.1304	0.1491	0.1468	0.1437	0.1439	0.1451	0.1546	0.1623	0.1537	0.1507	0.1503	0.154
strength of	0.05	0.1098	0.1292	0.1261	0.126	0.1258	0.1269	0.1248	0.1381	0.1362	0.1345	0.1347	0.1351
association β_1	0.5	0.1075	0.1207	0.1183	0.1182	0.1181	0.1195	0.13	0.143	0.1417	0.1425	0.1429	0.1425
missing proportion	60%	0.1091	0.1646	0.1387	0.1342	0.134	0.1342	0.129	0.2312	0.179	0.1849	0.1848	0.1723
		Binary complete (Z_b)						Continuous complete (Z_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		0.0997	0.1085	0.1023	0.1002	0.1003	0.1005	0.1375	0.1372	0.1393	0.1383	0.1384	0.1395
sample	200	0.3319	0.3477	0.3362	0.3367	0.3367	0.3354	0.4645	0.4927	0.4613	0.4568	0.4568	0.4592
size	500	0.1953	0.2168	0.2007	0.2013	0.2013	0.2005	0.3065	0.335	0.3043	0.3043	0.3043	0.3049
corr between	0.01	0.0899	0.0966	0.0894	0.0889	0.0889	0.0891	0.1314	0.1331	0.1309	0.1307	0.1307	0.1308
covariates (ρ)	0.5	0.1257	0.1378	0.1249	0.1264	0.1265	0.1249	0.1739	0.1817	0.1785	0.1763	0.1762	0.1746
strength of	0.05	0.102	0.1114	0.1031	0.1019	0.1019	0.1022	0.1379	0.1394	0.1372	0.1377	0.1378	0.1394
association β_1	0.5	0.095	0.1016	0.0994	0.0966	0.0966	0.0965	0.1414	0.1418	0.1412	0.1404	0.1405	0.1417
missing proportion (prop)	60%	0.0997	0.1599	0.1171	0.1033	0.1031	0.1048	0.1375	0.2179	0.1496	0.1388	0.1386	0.137

CD: estimators in complete data, CCA: complete cases analysis, MID: missing indicator, MI: MICE,
MI-Int: MICE with interaction of complete variables Z and hazard $H_0(\hat{T})$
When each factor is varied, the results of the baseline need to be added for comparison.
The number of replications is 100. In baseline model, sample size is 2000; correlations between the covariates are equal to 0.2; $\beta_1 = 0.2$;missing proportion is 20%

Table 3: Bias of the estimated coefficients (comparing with $\beta = \beta_1, \beta_2 = 0.5$) for MAR-T.

		Binary missing (X_b)						Continuous missing (X_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		-0.0036	-0.0263	-0.0321	-0.0119	-0.0117	-0.0009	-0.0033	-0.0263	-0.0466	-0.0379	-0.0376	-0.014
sample	200	0.0182	-0.0055	-0.0084	0.0098	0.0098	0.026	0.0356	0.037	0.0157	0.0204	0.0204	0.0487
	500	0.0306	0.014	0.0084	0.0187	0.0187	0.0306	-0.0268	-0.0445	-0.0576	-0.0587	-0.0587	-0.031
corr between	0.01	-0.0078	-0.0244	-0.0158	-0.0076	-0.0077	-0.0062	-0.0088	-0.0031	-0.0064	-0.0063	-0.0066	-0.0038
covariates	0.5	0.0029	0.0023	-0.0447	0.0203	0.0199	0.0206	0.0028	-0.0196	-0.0361	0.0308	0.0309	-0.0168
strength of	0.05	-0.0046	-0.02	-0.0366	-0.0151	-0.015	-0.0085	-0.0031	-0.005	-0.0258	-0.019	-0.019	0.0023
	0.5	0.0046	-0.0185	-0.0368	-0.0195	-0.0197	-0.0002	-0.001	-0.0217	-0.0415	-0.0335	-0.033	0.0013
missing proportion	60%	-0.0036	-0.0344	-0.0687	-0.0302	-0.0306	-0.0549	-0.0033	-0.0009	-0.0666	-0.0528	-0.0526	-0.0225
		Binary complete (Z_b)						Continuous complete (Z_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		0.0062	0.0816	0.1276	0.0113	0.0113	0.0096	-0.0074	0.1305	0.1559	-0.0017	-0.0019	-0.0063
sample	200	-0.0335	0.0183	0.0905	-0.0288	-0.0288	-0.0315	-0.0182	0.1001	0.143	-0.009	-0.009	-0.0184
	500	0.0043	0.0951	0.1158	0.0094	0.0094	0.0071	-0.0193	0.1315	0.1513	-0.0076	-0.0076	-0.0116
corr between	0.01	0.0076	0.0967	0.1024	0.0074	0.0073	0.0093	-0.0058	0.1111	0.1184	-0.0053	-0.0053	-0.0045
covariates (ρ)	0.5	-0.0109	0.0613	0.1102	-0.0224	-0.0224	-0.0052	-0.0055	0.0922	0.1682	-0.0249	-0.0247	-0.0063
strength of	0.05	0.006	0.0776	0.1175	0.0089	0.0089	0.0074	-0.0083	0.1097	0.1465	-0.0026	-0.0027	-0.0067
	0.5	-0.0005	0.0798	0.1254	0.0024	0.0024	0.0003	-0.0155	0.0897	0.1549	-0.0088	-0.0088	-0.0135
missing proportion (prop)	60%	0.0062	0.0732	0.156	0.0176	0.0175	0.034	-0.0074	0.1308	0.1981	0.0064	0.0063	0.0261

CD: estimators in complete data, CCA: complete cases analysis, MID: missing indicator, MI: MICE,
MI-Int: MICE with interaction of complete variables Z and hazard $H_0(\hat{T})$
When each factor is varied, the results of the baseline need to be added for comparison.
The number of replications is 100. In baseline model, sample size is 2000; correlations between the covariates are equal to 0.2; $\beta_1 = 0.2$;missing proportion is 20%

Table 4: RMSE of the estimated coefficients (comparing with $\beta = \beta_1, \beta_2 = 0.5$) for MAR-T.

		Binary missing (X_b)						Continuous missing (X_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		0.1091	0.1141	0.1122	0.1144	0.1143	0.1137	0.129	0.1592	0.1525	0.1463	0.1468	0.1489
sample	200	0.356	0.397	0.3505	0.3667	0.3667	0.3622	0.4751	0.5731	0.52	0.5115	0.5115	0.53
	500	0.2238	0.2556	0.2414	0.2512	0.2512	0.2537	0.2646	0.3008	0.2893	0.282	0.282	0.2866
corr between	0.01	0.104	0.1225	0.1119	0.1134	0.1133	0.1118	0.1213	0.1516	0.1424	0.1408	0.141	0.147
covariates	0.5	0.1304	0.1486	0.1413	0.145	0.1448	0.1461	0.1546	0.1843	0.1714	0.1704	0.1713	0.1761
strength of	0.05	0.1098	0.1223	0.125	0.1228	0.1227	0.1214	0.1248	0.1519	0.1409	0.1404	0.1404	0.1447
	0.5	0.1075	0.125	0.1147	0.1151	0.1149	0.1144	0.13	0.1517	0.1418	0.1405	0.1401	0.1422
missing proportion	60%	0.1091	0.189	0.157	0.1569	0.1564	0.1463	0.129	0.2229	0.1825	0.1749	0.1754	0.1638
		Binary complete (Z_b)						Continuous complete (Z_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		0.0997	0.1455	0.1647	0.101	0.1009	0.1013	0.1375	0.2012	0.2098	0.1403	0.1404	0.1408
sample	200	0.3319	0.3842	0.3401	0.3373	0.3373	0.3376	0.4645	0.5558	0.511	0.4611	0.4611	0.4654
	500	0.1953	0.235	0.23	0.1986	0.1986	0.2003	0.3065	0.3475	0.3493	0.3121	0.3121	0.3113
corr between	0.01	0.0899	0.143	0.1387	0.0904	0.0904	0.0908	0.1314	0.1837	0.1798	0.1302	0.1301	0.1308
covariates (ρ)	0.5	0.1257	0.1507	0.1669	0.1312	0.1312	0.1286	0.1739	0.2111	0.2382	0.1789	0.179	0.1753
strength of	0.05	0.102	0.1381	0.1542	0.1028	0.1027	0.1035	0.1379	0.1931	0.2045	0.1396	0.1395	0.1409
	0.5	0.095	0.1379	0.1575	0.0951	0.0952	0.0952	0.1414	0.1838	0.2078	0.1417	0.1415	0.1421
missing proportion (prop)	60%	0.0997	0.2073	0.1846	0.102	0.1022	0.1045	0.1375	0.263	0.2401	0.1474	0.1476	0.1448

CD: estimators in complete data, CCA: complete cases analysis, MID: missing indicator, MI: MICE,
MI-Int: MICE with interaction of complete variables Z and hazard $H_0(\hat{T})$
When each factor is varied, the results of the baseline need to be added for comparison.
The number of replications is 100. In baseline model, sample size is 2000; correlations between the covariates are equal to 0.2; $\beta_1 = 0.2$;missing proportion is 20%

4 Application

This chapter will analyse a real case study. The original aim of this case was to compare philadelphia+ chronic myelogenous leukemia (Ph+ CML) patients candidate for allogeneic hematopoietic cell transplantation (allo-HCT) from a haploidentical (haplo) donor for Ph+ CML might be advantageous in compar- ison to matched related donor (MRD), matched unrelated donor (MUD) or mismatched unrelated donor (MMUD). For the purpose of this article, we will focus only on the overall survival endpoint.

4.1 Clinical background

Chronic myeloid leukaemia (CML) is a type of leukaemia (commonly known as blood cancer) that causes abnormal proliferation of white blood cells and usually develops slowly, over a period of months or even six months, before becoming life-threatening. The disease is caused by a genetic mutation in the blood-forming cells of the bone marrow (BM). This genetic mutation can be detected in about 90-95% of patients with chronic myeloid leukaemia.

The use of tyrosine kinase inhibitors (TKI) was introduced in clinical practice in 2001. However, the number of people treated with allo-HCT for CML has remained stable at around 300 per year in recent years in Europe. Although TKI reduces the frequency of advanced disease, it does not perform well at the end stage of the disease. Based on previous studies, a combination of both modalities is recommended for treatment. Thus, in the first chronic phase (CP1), TKI resistance was the most common transplant indication for allogeneic haematological stem cell transplantation (allo-HSCT).

In recent years, the use of haploidentical donors (Haplo/MMRD) has gradually increased thanks to improved protocols for complete T-cell transplantation, and its transplantation results are comparable to those of matched related donor (MRD), matched unrelated donor (MUD) or mismatched unrelated donor (MMUD). It seems that for high-risk patients, Haplo is even more beneficial.

4.2 Data description

To explore whether Haplo has an advantage, the following covariates were included in the analysis when estimating hazard ratios: X_{age} refers to the age of patients, X_{don} refers to the donor type, X_{ks} refers to Karnofsky score (This runs from 100 to 0, where 100 was "perfect" health and 0 was death.), X_{cmv} refers to Cytomegalovirus in patient (2 classes are $+$ and $-$), X_{ric} refers to reduced intensity conditioning (2 classes are reduced and standard), X_{stage} refers to disease status (4 classes are CP1, CP2 or more, accelerated phase (AP) and blast crisis (BC)) and X_{ss} refers to source of the stem cell (bone marrow (BM) or peripheral blood (PB)). The total number of patients in this dataset was 1686, of which 534 (31.7%) patients died and 171 (10.1%) patients were censored. The mean age of the patients at transplantation was 45 years. Table 3 shows the descriptive statistics and the correlation matrix for covariates. Among them, X_{age} and X_{don} are complete covariates. As most of these variables are categorical and have missing values, the correlation coefficients here are calculated using Spearman's correlation coefficients for pairwise complete observations which are 1482 patients.

Table 5: Descriptive analysis of covariates

	mean	**median**	**[min, max]**	$N_{missing}$				
X_{age}	45.2	46.1	[18.0,73.7]	0				
	class	**size (% among N)**	$N_{missing}$			**class**	**size (% among N)**	$N_{missing}$
X_{don}	HD	136 (8.1)	0		X_{ric}	reduced	552 (32.7)	24 (1.4)
	MRD	661 (39.2)				standard	1110 (65.8)	
	MUD	677 (40.2)			X_{stage}	CP1	718 (42.6)	33 (2.0)
	MMUD	212 (12.6)				CP2 or more	445 (26.4)	
X_{ks}	>=90	1240 (73.6)	104 (6.2)			AP	194 (11.5)	
	<90	342 (20.3)				BC	296 (17.6)	
X_{cmv}	+	1062 (63.0)	68 (4.0)		X_{ss}	PB	1418 (84.1)	8 (0.5)
	-	556 (33.0)				BM	260 (15.4)	

PB: peripheral blood. *BM:* bone marrow. *CP:* chronic phase,1 or 2 means the number of chronic phase. *AP:* accelerated phase. *BC:*blast crisis also called blast phase.

Table 6: Correlation matrix

	X_{age}	X_{don}	X_{ks}	X_{cmv}	X_{ric}	X_{stage}	X_{ss}
X_{age}	1.0000	0.0828	0.0702	-0.0056	**0.4213**	-0.0256	-0.1229
X_{don}	0.0828	1.0000	-0.0183	-0.1366	0.0813	-0.0958	-0.1530
X_{ks}	0.0702	-0.0183	1.0000	0.0490	0.0911	0.1259	-0.0058
X_{cmv}	-0.0056	-0.1366	0.0490	1.0000	-0.0550	0.0890	0.0363
X_{ric}	**0.4213**	0.0813	0.0911	-0.0550	1.0000	-0.0413	-0.1170
X_{stage}	-0.0256	-0.0958	0.1259	0.0890	-0.0413	1.0000	-0.0363
X_{ss}	-0.1229	-0.1530	-0.0058	0.0363	-0.1170	-0.0363	1.0000

It is clear to see that most of the variables are weakly correlated with each other, except for X_{ric} and X_{age} which are weakly correlated.

4.3 Analysis

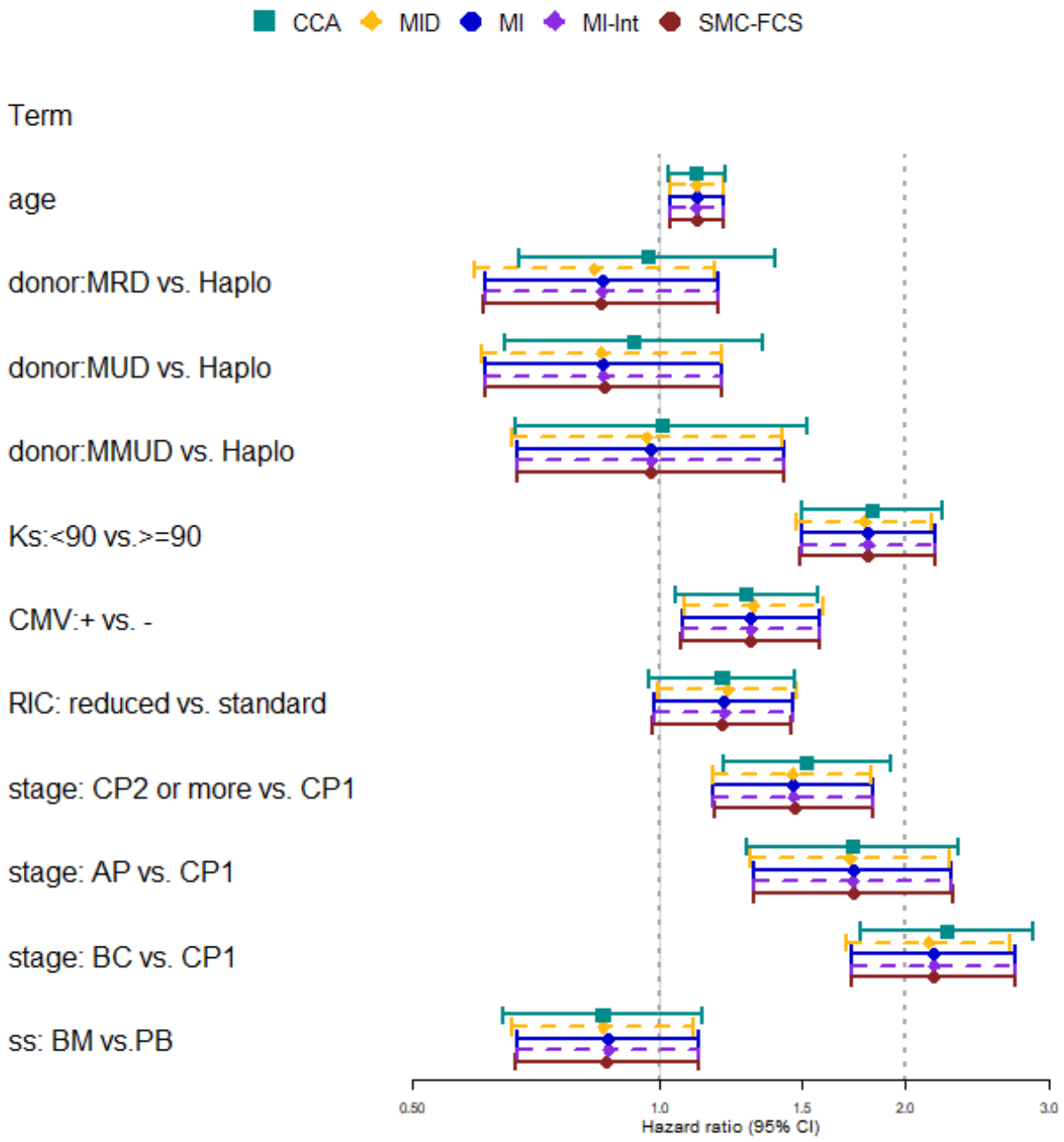


Figure 8: Forest plot with estimated HR and 95% CI for the Cox model. On the x-axis are the hazard ratios. (need to change to plot: on the log scale where the confidence intervals are symmetric.)

The choice of covariates for the above model was determined based on univariate analysis as well as the research context of the subject. The proportional hazard assumptions of the Cox model were verified to be satisfied. Unlike in the simulation experiments, there were covariates not in the model that were added as one of the complete variables during the imputation process. For example, the year in which the observations were collected. Observations from earlier years may have more missing values due to improvements to the way data are collected. In addition, to ensure that the iterative process in smcfcs converges, N_{iter} is raised here to 5. No non-convergence was observed through the observations. In the present data, the only variables with missing values are multicategorical and dichotomous variables. Thus, dichotomous covariates were imputed using logistic regression, and unordered categorical variables were imputed using multinomial logistic regression. Figure 8 summarises the estimated HRs which is the exponential of coefficients, and associated 95% confidence intervals (CI), which are based on the pooled standard errors and the t-distribution.

It could be seen that the CI of CCA is slightly wider than the other methods. CCA produces larger differences from the other methods in the estimation results for X_{don}, X_{ks} and X_{stage}. However, this difference is not in fact on the variables as a whole. Similarly, for the estimation of disease status, the apparent difference is between the categories BC and CP2 or more, and by looking at the distribution of the number of categories in the descriptive statistics, it is possible to speculate that one of the reasons for this may be an imbalance in the proportion of the number of categories. For example, the number of haplo in the donor type is low and the information is already limited. By looking at the classification of observations with missing values in the donor type it can be found that about 10% of haplo need to be excluded from the complete case analysis, again reducing the information in this category. The difference in performance between MICE and SMC-FCS is not very large. A similar situation is found in Bartlett's article [20].

5 Discussion

In this paper, we evaluate the performance of complete cases analysis, missing indicator, MICE and SMC-FCS for processing Cox models with missing data. Based on the characteristics of the data in the real case studies, we consider categorical variables as well as continuous variables. Although there is no missing values in the continuous variable we used, it has many missing values in other continuous variables from the real data set, like age of donors. This is a result of the analysis in a specific data structure. In the simulation study, we considered a variety of scenarios, which covered parameters such as the type of missingness, and the strength of association. In each scenario, we modified only one variable each time. For the missing data, we emphasise that the overall missingness mechanism is of one type. For the analysis of the results, we have focused on bias and RMSE.

The simulation results show that although CCA theoretically loses a lot of data information, it can perform very well in the MCAR missing scenario and that it generally performs well for parameter estimation of categorical variables, whereas MID has very limited applications and even performs poorly in MAR. We consider that MID produces a large bias by adding new variables in order to express the missing case of one variable, which has some impact on the nature of the model. When missing variables are correlated with other variables, the new variables introduced thus also have an impact on the other variables. At the same time, the missing proportion of baseline covariates is not balanced across different groups is also one of the reason that missing indicator causes bias results. Both SMC-FCS and MICE also show unbiased performance in MCAR as well as MAR scenarios. In general SMC-FCS outperforms MICE, especially when the correlation between the variables is strong. To improve performance, MICE can be extended to include and interaction between covariates and the cumulative hazard. However, this extension is not lead to significant different results compared to the MICE imputation model without interaction in the scenarios studied. As the proportion of missing data increases, the performance gap between MICE and SMC-FCS becomes larger. From the imputation principles of MICE and SMC-FCS, it is clear that they work

mainly in the context of MAR missingness. However, when the missingness is correlated with survival time MAR-T, all methods show more bias than normal MAR. Besides, surprisingly MICE and SMC-FCS produce even more bias than CCA, like in MNAR missingness mechanism.

There are some limitations to this paper. The first is in the setting of the scenarios. In addition to dichotomous and continuous variables, ordered categorical variables and multi-categorical variables could have been included in the discussion. In the process of generating missing data, many adjustable parameters can be found, and there are many more complex missing data scenarios discussed for study, for example, some data are MCAR and some are MAR. if one wants to have multiple missing mechanisms in the same dataset, this can be achieved using adjustments to the pattern and corresponding weight score. For example, weak MAR is composed of part MCAR and part MAR, as discussed in Schouten's paper [18]. These more complex scenarios could be studied in a new study. Secondly, the number of iterations of SMC-FCS was set relatively low in order to speed up the experimental time. This resulted in very few cases of non-convergence of results during the experiment. At the same time, the sample sizes in this thesis are small in the generated data. If we check the parameter estimation results based on the complete data (CD) it is also biased sometimes. Consider the fact that there are many other endpoints such as Relapse and NRM in the actual clinical analysis. Further sub-exploration could also include modelling with other endpoints. In this article, the inclusion of interaction terms in MICE did not have a significant impact on performance.However, in the results of White and Royston's study[4], the two approaches are different. Moreover, for time saving, we only use 100 replications in Monte Carlo simulations. If the number of replications could be increased to 1000, the results might be more reliable.

The aim of our study is to compare which method is better in different scenarios. Looking at the results of the simulation experiments together, SMC-FCS is the most robust approach. It gave more stable results when there was more missing data and complex relationships between variables. Although we spent a lot of time running SMC-FCS during the simulation study, it took about four times as long as

MICE, this is due to the Monte Carlo replications. In the real case, the imputation only needs to be done once. The time consuming problem can be ignored unless the size of the data set is large and the imputation parameters are set relatively large. MID can easily produce results with large bias in similar data sets. The simplest implementation CCA may produce unbiased results in many scenarios, such as MNAR missingness and when there is relatively little missing data. Combined with the results of the previous section on real data, CCA and SMC-FCS can be applied in combination, and take into account the confidence intervals of the coefficients estimated by these two methods.

References

[1] Rolf HH Groenwold, Ian R White, A Rogier T Donders, James R Carpenter, Douglas G Altman, and Karel GM Moons. Missing covariate data in clinical research: when and when not to use the missing-indicator method for analysis. *Cmaj*, 184(11):1265–1269, 2012.

[2] A Rogier T Donders, Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091, 2006.

[3] Ian R White and Simon G Thompson. Adjusting for partially missing baseline measurements in randomized trials. *Statistics in medicine*, 24(7):993–1007, 2005.

[4] Ian R White and Patrick Royston. Imputing missing covariate values for the cox model. *Statistics in medicine*, 28(15):1982–1998, 2009.

[5] Jonathan W Bartlett, Shaun R Seaman, Ian R White, James R Carpenter, and Alzheimer's Disease Neuroimaging Initiative*. Multiple imputation of covariates by fully conditional specification: accommodating the substantive model. *Statistical methods in medical research*, 24(4):462–487, 2015.

[6] Terry M Therneau and Patricia M Grambsch. The cox model. In *Modeling survival data: extending the Cox model*, pages 39–77. Springer, 2000.

[7] Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[8] Joseph L Schafer. *Analysis of incomplete multivariate data*. CRC press, 1997.

[9] James Carpenter and Michael Kenward. *Multiple imputation and its application*. John Wiley & Sons, 2012.

[10] Jingchen Liu, Andrew Gelman, Jennifer Hill, Yu-Sung Su, and Jonathan Kropko. On the stationary distribution of iterative imputations. *Biometrika*, 101(1):155–173, 2014.

[11] Jonathan W Bartlett and Tim P Morris. Multiple imputation of covariates by substantive-model compatible fully conditional specification. *The Stata Journal*, 15(2):437–456, 2015.

[12] R Core Team et al. R: A language and environment for statistical computing. 2013.

[13] Jonathan Kropko and Jeffrey J Harden. Beyond the hazard ratio: generating expected durations from the cox proportional hazards model. *British Journal of Political Science*, 50(1):303–320, 2020.

[14] Terry M Therneau and Thomas Lumley. Package 'survival'. *Survival analysis Published on CRAN*, 2(3):119, 2014.

[15] Stef Van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 45(1):1–67, 2011.

[16] Hakan Demirtas and Beyza Doganay. Simultaneous generation of binary and normal data with specified marginal and association structures. *Journal of Biopharmaceutical Statistics*, 22(2):223–236, 2012.

[17] Hakan Demirtas and Donald Hedeker. Computing the point-biserial correlation under any underlying continuous distribution. *Communications in Statistics-Simulation and Computation*, 45(8):2744–2751, 2016.

[18] Rianne Margaretha Schouten, Peter Lugtig, and Gerko Vink. Generating missing values for simulation purposes: a multivariate amputation procedure. *Journal of Statistical Computation and Simulation*, 88(15):2909–2930, 2018.

[19] Tim P Morris, Ian R White, and Michael J Crowther. Using simulation studies to evaluate statistical methods. *Statistics in medicine*, 38(11):2074–2102, 2019.

[20] Jonathan W Bartlett and Jeremy MG Taylor. Missing covariates in competing risks analysis. *Biostatistics*, 17(4):751–763, 2016.

Appendix

A. MLE of Weibull distribution parameters

Weibull Cumulative distribution function (CDF) and Probability density function (PDF) are as following with shape parameter α and scale parameter β:

$$F(x) = 1 - e^{-(x/\beta)^{\alpha}},$$
$$f(x) = \frac{\alpha}{x}\gamma^{\alpha}e^{-\gamma^{x}}$$

here $\gamma = \alpha/\beta$. Definition of censored likelihood function is:

$$L = \prod_{i=1}^{n}(f(x_i))^{\delta_i}(1 - F(x_i))^{1-\delta_i},$$
$$\delta_i = \begin{cases} 1 & \text{if } x \leq threshold, \\ 0 & \text{if } x > threshold. \end{cases}$$

Substitute the CDF and PDF into both sides of the likelihood function to find the logarithm, and simplify to find the log-likelihood function:

$$log_L = \sum_{i=1}^{n}(-\gamma_i^{\alpha} + \delta_i \alpha ln\gamma_i + \delta ln\frac{\alpha}{x_i})$$

In order to maximise this natural function for the parametric expression, find the first order derivative of the above equation and make it zero:

$$\sum_{i=1}^{n}\gamma_i^{\alpha} - \delta_i = 0 \quad \Rightarrow \beta = (\frac{\sum_i^n x_i^{\alpha}}{\sum_i^n \delta_i})^{\frac{1}{\alpha}},$$
$$\sum_{i=1}^{n}(-\gamma_i^{\alpha} ln\gamma_i + \delta_i ln\gamma_i + \frac{\delta_i}{\alpha}) = 0,$$

Then we assume $h(\alpha) = \sum_{i=1}^{n}(-\gamma_i^{\alpha} ln\gamma_i + \delta_i ln\gamma_i + \frac{\delta_i}{\alpha})$ and with using Newton-Rapson method:

$$h(\alpha) = n(-\frac{\sum x_i^{\alpha} lnx_i}{\sum x_i^{\alpha}} + \frac{1}{\alpha} + \frac{1}{n}\sum \delta_i lnx_i),$$
$$h'(\alpha) = n(-\frac{1}{\alpha^2} + \frac{\sum x_i^{a}(lnx_i)^{\alpha}}{\sum x_i^{\alpha}} - \frac{(\sum x_i^{\alpha} lnx_i)^2}{(\sum x_i^{\alpha})^2})$$

Using Newton iterative method, the following equation can be solved for $h(\alpha) = 0$ by iterating over the following formula. Then α and β could be find step by step.

$$\alpha_{k+1} = \alpha_k - \frac{h(\alpha_k)}{h'(\alpha_k)}$$

B. Bias and RMSE tables

Table 7: Bias of the estimated coefficients (comparing with $\beta = \beta_1, \beta_2$) for MNAR.

		Binary missing (X_b)						Continuous missing (X_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		-0.0036	-0.0029	-0.0172	-0.0161	-0.0161	-0.012	-0.0033	0.0012	-0.0165	-0.0176	-0.0172	0.0004
sample	200	0.0182	0.0362	0.0225	0.0181	0.0181	0.0292	0.0356	0.093	0.0489	0.0412	0.0412	0.0534
size	500	0.0306	0.0061	0.0109	0.0135	0.0135	0.018	-0.0268	-0.0374	-0.0544	-0.058	-0.058	-0.0431
corr between	0.01	-0.0078	-0.0076	-0.0116	-0.0142	-0.0139	-0.0136	-0.0088	-0.017	-0.013	-0.0153	-0.0154	-0.008
covariates	0.5	0.0029	0.0053	-0.0254	0.0051	0.0049	-0.0082	0.0028	-0.017	-0.017	0.0307	0.0305	-0.0102
strength of	0.05	-0.0046	-0.0132	-0.0112	-0.0094	-0.0091	-0.0099	-0.0031	-0.0023	-0.0102	-0.0118	-0.0118	0.0013
association β_1	0.5	0.0046	0.01	-0.011	-0.0183	-0.0185	-0.0028	-0.001	0.0059	-0.0229	-0.0285	-0.0283	-0.0039
missing proportion	60%	-0.0036	0.0113	-0.0215	-0.019	-0.0189	-0.037	-0.0033	0.0028	-0.0336	-0.0424	-0.0422	-0.0199
		Binary complete (Z_b)						Continuous complete (Z_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		0.0062	0.0019	0.0299	0.0205	0.0205	0.0188	-0.0074	-0.0062	0.0279	0.0148	0.0147	0.012
sample	200	-0.0335	-0.0464	-0.0002	-0.022	-0.022	-0.0244	-0.0182	-0.0064	0.0356	0.0039	0.0039	0.0011
size	500	0.0043	0.0066	0.0288	0.021	0.021	0.0205	-0.0193	-0.0256	0.0183	0.0075	0.0075	0.0058
corr between	0.01	0.0076	0.0074	0.0081	0.0069	0.0069	0.0077	-0.0058	-0.0005	-0.0032	-0.0027	-0.0026	-0.0028
covariates (ρ)	0.5	-0.0109	-0.0091	0.03	-0.0029	-0.0028	0.0118	-0.0055	0.0031	0.0606	-0.0008	-0.0007	0.0188
strength of	0.05	0.006	0.0094	0.0238	0.0178	0.0177	0.0174	-0.0083	-0.0083	0.0203	0.0115	0.0114	0.01
association β_1	0.5	-0.0005	0.0022	0.0303	0.019	0.019	0.0156	-0.0155	-0.0273	0.0254	0.012	0.012	0.007
missing proportion (prop)	60%	0.0062	0.0156	0.0652	0.0269	0.0269	0.0372	-0.0074	0.0235	0.0779	0.0254	0.0255	0.039

Table 8: Bias of the estimated coefficients (comparing with $\beta = \beta_1, \beta_2$) for MAR.

		Binary missing (X_b)						Continuous missing (X_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		-0.0036	0.0005	-0.0159	-0.0159	-0.0163	-0.0081	-0.0033	-0.0035	-0.0228	-0.0298	-0.0298	-0.0096
sample	200	0.0182	0.0366	0.0109	0.0045	0.0045	0.0103	0.0356	0.0422	0.0299	0.0119	0.0119	0.0304
size	500	0.0306	0.0135	0.0139	0.0082	0.0082	0.0171	-0.0268	-0.0385	-0.0516	-0.06	-0.06	-0.0421
corr between	0.01	-0.0078	-0.0039	-0.0078	-0.0123	-0.0124	-0.0097	-0.0088	-0.0079	-0.0066	-0.0115	-0.0112	-0.0072
covariates	0.5	0.0029	0.0097	-0.0253	0.0092	0.0088	-0.0021	0.0028	0.0051	0.0069	0.0442	0.0441	-0.0053
strength of	0.05	-0.0046	-0.0046	-0.0093	-0.0083	-0.0081	-0.0065	-0.0031	-0.0112	-0.0167	-0.0195	-0.0195	-0.0052
association β_1	0.5	0.0046	0.0128	-0.0034	-0.0092	-0.0092	0.0078	-0.001	0.013	-0.0172	-0.0276	-0.028	-0.0021
missing proportion	60%	-0.0036	-0.02	-0.0407	-0.0322	-0.0314	-0.0591	-0.0033	-0.0048	-0.0375	-0.049	-0.049	-0.035
		Binary complete (Z_b)						Continuous complete (Z_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		0.0062	0.0138	0.03	0.0099	0.01	0.0093	-0.0074	-0.0057	0.0269	-0.0029	-0.0028	-0.0046
sample	200	-0.0335	-0.0915	-0.0377	-0.0308	-0.0308	-0.0307	-0.0182	-0.0719	-0.0121	-0.0037	-0.0037	-0.0021
size	500	0.0043	0.0216	0.039	0.0122	0.0122	0.0131	-0.0193	0.0297	0.0368	-0.0067	-0.0067	-0.0069
corr between	0.01	0.0076	0.01	0.0084	0.0064	0.0064	0.0079	-0.0058	-0.0058	-0.0052	-0.0072	-0.0072	-0.0057
covariates (ρ)	0.5	-0.0109	-0.0132	0.0181	-0.0229	-0.0228	-0.0013	-0.0055	-0.0076	0.0613	-0.0226	-0.0224	0.0028
strength of	0.05	0.006	0.0212	0.028	0.0086	0.0086	0.009	-0.0083	0.0068	0.0247	-0.0048	-0.0048	-0.0048
association β_1	0.5	-0.0005	-0.0164	0.0273	0.0025	0.0026	0.0021	-0.0155	-0.0133	0.0275	-0.0092	-0.009	-0.0108
missing proportion (prop)	60%	0.0062	0.0029	0.067	0.0154	0.0152	0.0359	-0.0074	-0.003	0.0783	0.0071	0.007	0.0309

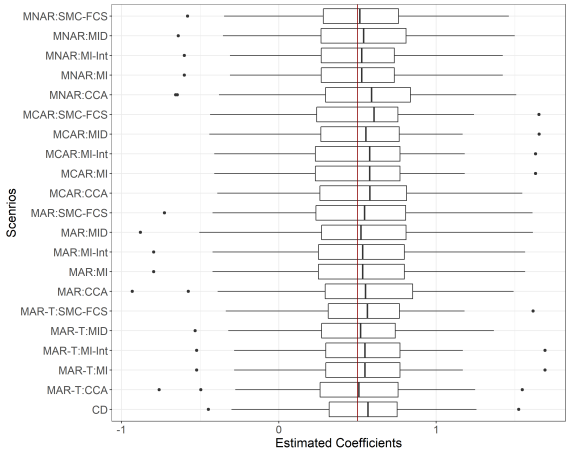
Table 9: RMSE of the estimated coefficients (comparing with $\beta = \beta_1, \beta_2$) for MNAR.

		Binary missing (X_b)						Continuous missing (X_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		0.1091	0.1317	0.1227	0.126	0.1256	0.1259	0.129	0.1556	0.133	0.1333	0.134	0.1344
sample	200	0.356	0.4408	0.3902	0.3819	0.3819	0.3915	0.4751	0.5413	0.5	0.4884	0.4884	0.5025
size	500	0.2238	0.2552	0.2465	0.2432	0.2432	0.2459	0.2646	0.3006	0.2875	0.2822	0.2822	0.2833
corr between	0.01	0.104	0.1172	0.1109	0.111	0.111	0.1116	0.1213	0.1427	0.1348	0.1382	0.1381	0.1348
covariates	0.5	0.1304	0.1558	0.1325	0.135	0.1349	0.1372	0.1546	0.1817	0.1573	0.1562	0.1562	0.1613
strength of	0.05	0.1098	0.129	0.1235	0.1233	0.1236	0.1228	0.1248	0.138	0.1322	0.1302	0.1304	0.1319
association β_1	0.5	0.1075	0.1246	0.1225	0.1236	0.1238	0.1244	0.13	0.1623	0.1368	0.1412	0.1415	0.141
missing proportion	60%	0.1091	0.1853	0.1493	0.1527	0.1532	0.1475	0.129	0.2077	0.1587	0.1514	0.1508	0.1415
		Binary complete (Z_b)						Continuous complete (Z_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		0.0997	0.1266	0.1018	0.1004	0.1004	0.1002	0.1375	0.165	0.1394	0.1381	0.1383	0.1382
sample	200	0.3319	0.4023	0.334	0.3321	0.3321	0.3368	0.4645	0.5288	0.4685	0.4615	0.4615	0.4648
size	500	0.1953	0.2299	0.2015	0.2008	0.2008	0.2009	0.3065	0.3563	0.305	0.3017	0.3017	0.3039
corr between	0.01	0.0899	0.0963	0.091	0.0899	0.0898	0.0899	0.1314	0.1527	0.1312	0.1315	0.1315	0.1316
covariates (ρ)	0.5	0.1257	0.1399	0.1273	0.1273	0.1271	0.1273	0.1739	0.1886	0.1764	0.1751	0.175	0.1738
strength of	0.05	0.102	0.1224	0.103	0.1053	0.1053	0.1057	0.1379	0.1553	0.1377	0.1395	0.1396	0.1392
association β_1	0.5	0.095	0.1085	0.0977	0.0965	0.0964	0.0961	0.1414	0.1825	0.1398	0.1418	0.1419	0.1418
missing proportion (prop)	60%	0.0997	0.1744	0.1168	0.1041	0.1041	0.1057	0.1375	0.218	0.1532	0.1393	0.1392	0.1415

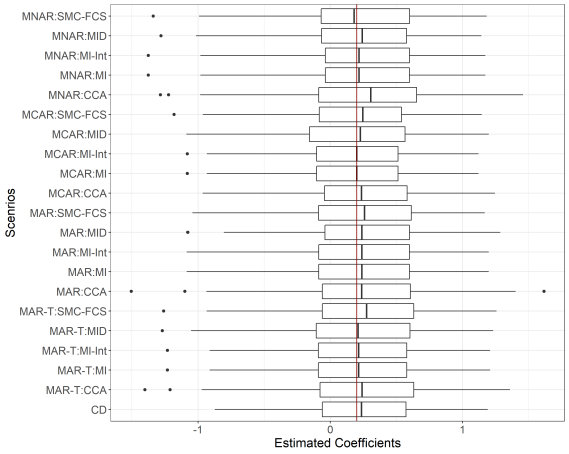
Table 10: RMSE of the estimated coefficients (comparing with $\beta = \beta_1, \beta_2$) for MAR.

		Binary missing (X_b)						Continuous missing (X_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		0.1091	0.1328	0.1206	0.1208	0.1206	0.1207	0.129	0.1413	0.1401	0.1383	0.138	0.1397
sample	200	0.356	0.4256	0.4115	0.3999	0.3999	0.4044	0.4751	0.5192	0.5083	0.4778	0.4778	0.485
size	500	0.2238	0.2975	0.2626	0.2552	0.2552	0.2599	0.2646	0.3052	0.287	0.2843	0.2843	0.2821
corr between	0.01	0.104	0.1115	0.1077	0.1072	0.1071	0.107	0.1213	0.1352	0.1275	0.1272	0.1273	0.1265
covariates	0.5	0.1304	0.1499	0.135	0.1318	0.1312	0.1353	0.1546	0.1772	0.1565	0.1649	0.1651	0.1648
strength of	0.05	0.1098	0.132	0.1224	0.1211	0.1212	0.1232	0.1248	0.1341	0.1327	0.1299	0.1298	0.134
association β_1	0.5	0.1075	0.1216	0.1208	0.1175	0.1173	0.1192	0.13	0.1514	0.1501	0.1484	0.1482	0.1518
missing proportion	60%	0.1091	0.2036	0.1569	0.1552	0.1554	0.1548	0.129	0.2011	0.1624	0.1614	0.1626	0.1515
		Binary complete (Z_b)						Continuous complete (Z_c)					
		CD	CCA	MID	MI	MI-Int	SMC-FCS	CD	CCA	MID	MI	MI-Int	SMC-FCS
baseline		0.0997	0.1172	0.1054	0.1012	0.1011	0.1008	0.1375	0.1675	0.1463	0.1391	0.1391	0.1395
sample	200	0.3319	0.4403	0.3475	0.3337	0.3337	0.3345	0.4645	0.5275	0.4896	0.4793	0.4793	0.4808
	500	0.1953	0.2455	0.2145	0.1986	0.1986	0.1984	0.3065	0.3652	0.3179	0.3042	0.3042	0.3076
corr between	0.01	0.0899	0.1084	0.0963	0.0904	0.0904	0.0907	0.1314	0.1319	0.1346	0.1317	0.1317	0.1318
covariates (ρ)	0.5	0.1257	0.1465	0.1251	0.1273	0.1272	0.1238	0.1739	0.1953	0.1827	0.1781	0.1781	0.1746
strength of	0.05	0.102	0.1257	0.1094	0.104	0.1039	0.1038	0.1379	0.1569	0.1401	0.1372	0.1372	0.1382
associationβ_1	0.5	0.095	0.1161	0.0996	0.0954	0.0953	0.095	0.1414	0.1557	0.14	0.1404	0.1404	0.1406
missing proportion (prop)	60%	0.0997	0.2047	0.1218	0.1049	0.1051	0.1082	0.1375	0.2318	0.1516	0.1432	0.1432	0.1429

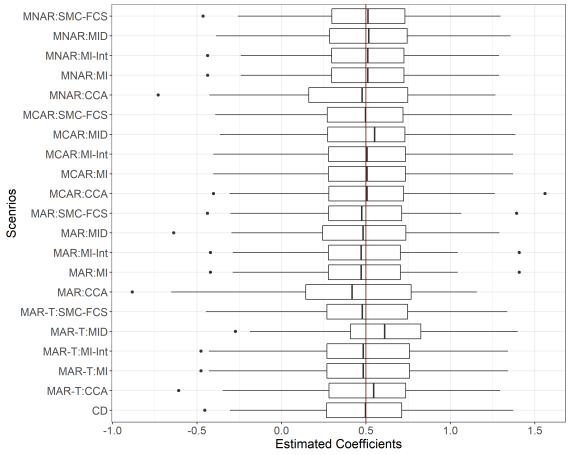
C. Box plots



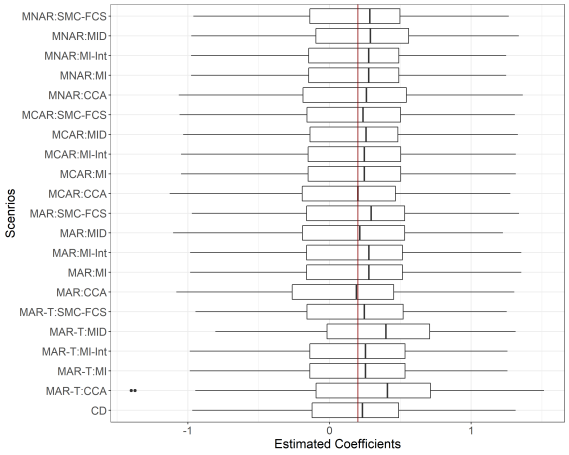
(a) Coefficients of X_b



(b) Coefficients of X_c

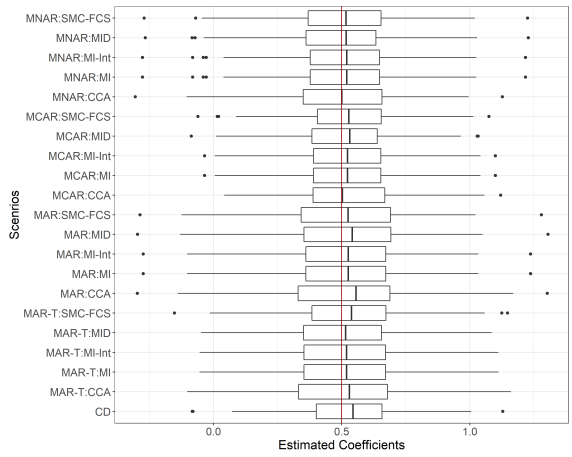


(c) Coefficients of Z_b

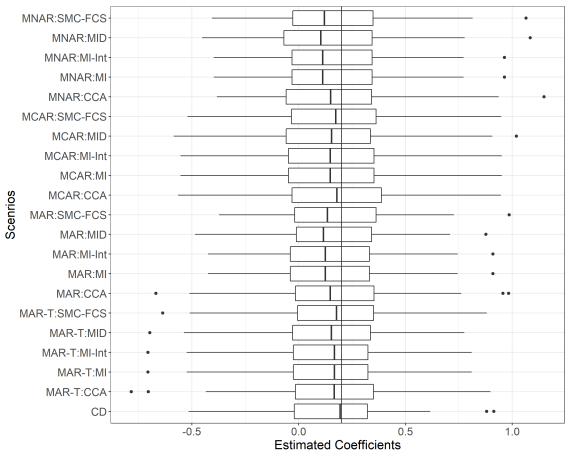


(d) Coefficients of Z_c

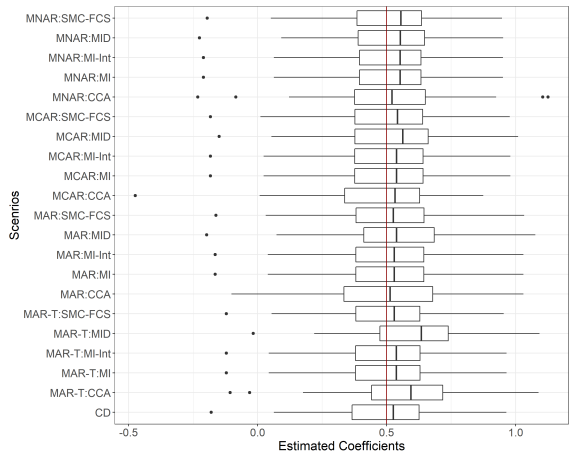
Figure 9: Box plots of Monte Carlo estimated coefficients in different scenarios when sample size is equal to 200.



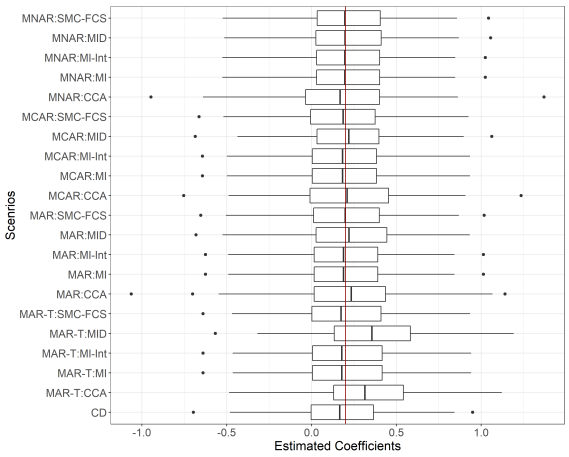
(a) Coefficients of X_b



(b) Coefficients of X_c

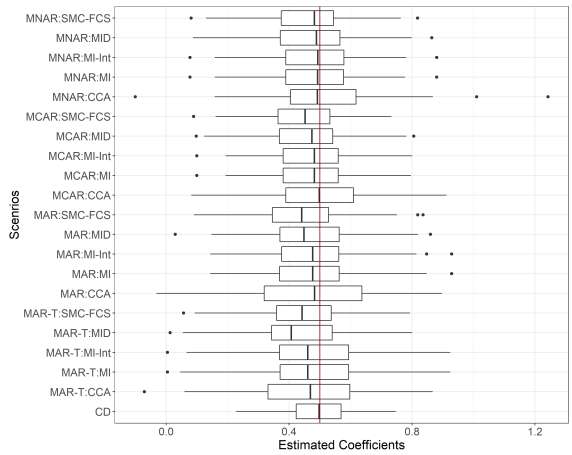


(c) Coefficients of Z_b

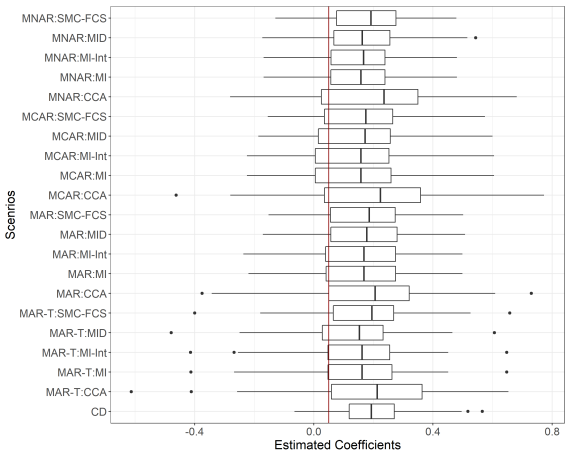


(d) Coefficients of Z_c

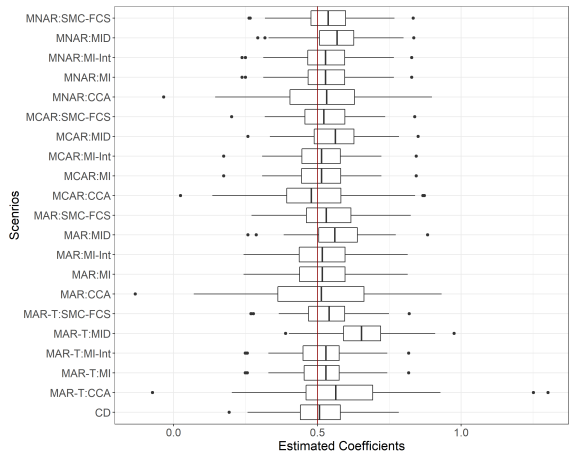
Figure 10: Box plots of Monte Carlo estimated coefficients in different scenarios when sample size is equal to 500.



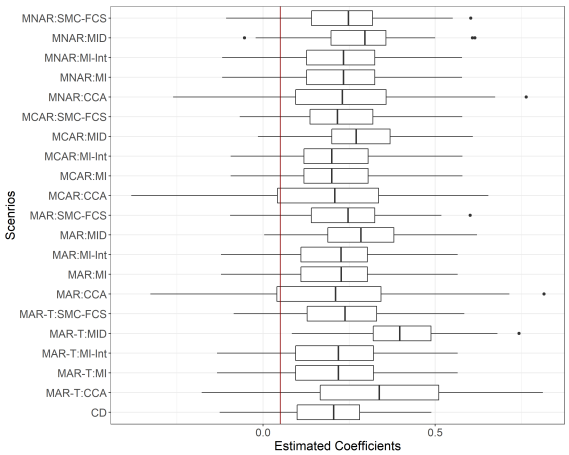
(a) Coefficients of X_b



(b) Coefficients of X_c

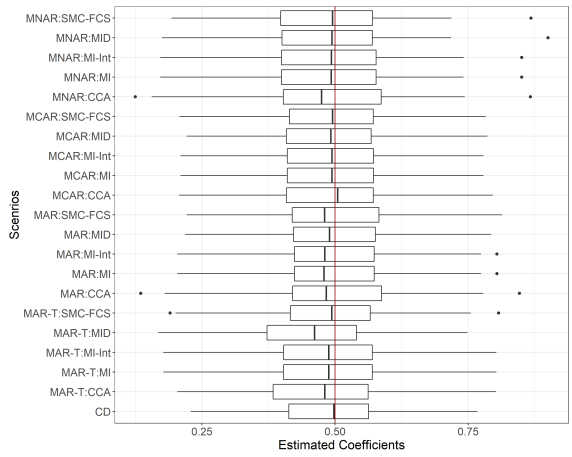


(c) Coefficients of Z_b

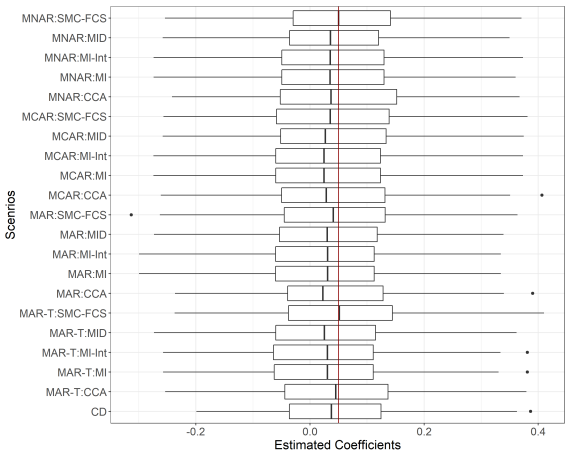


(d) Coefficients of Z_c

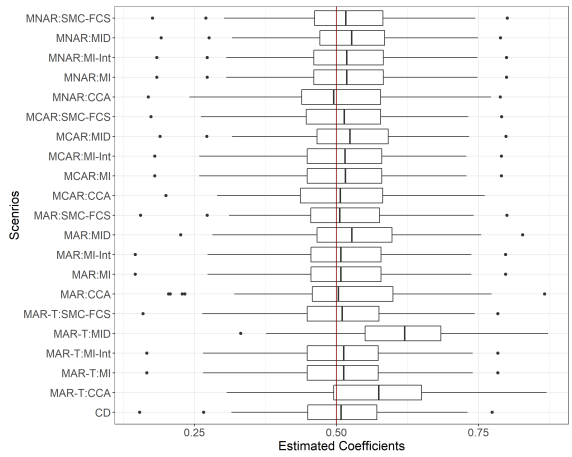
Figure 11: Box plots of Monte Carlo estimated coefficients in different scenarios when 60% data is missing.



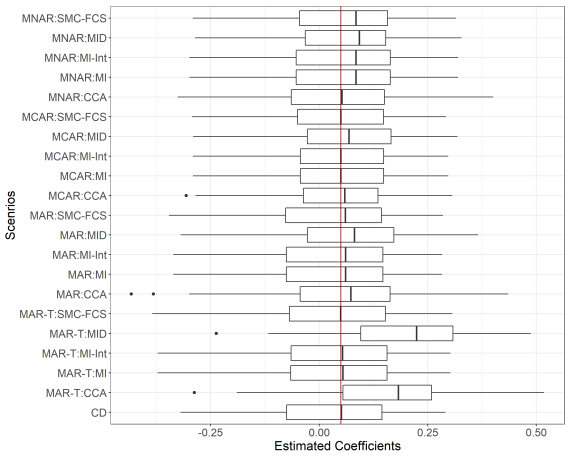
(a) Coefficients of X_b



(b) Coefficients of X_c

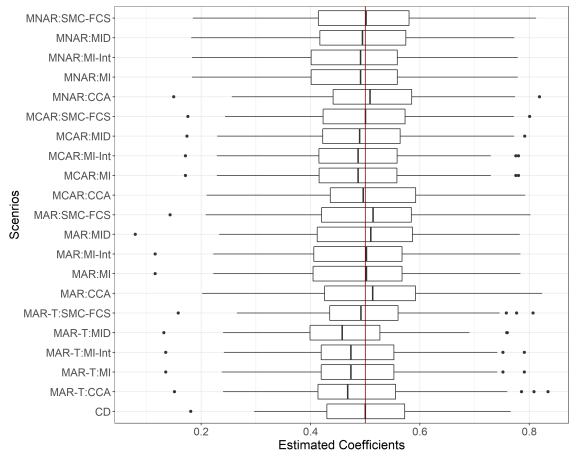


(c) Coefficients of Z_b

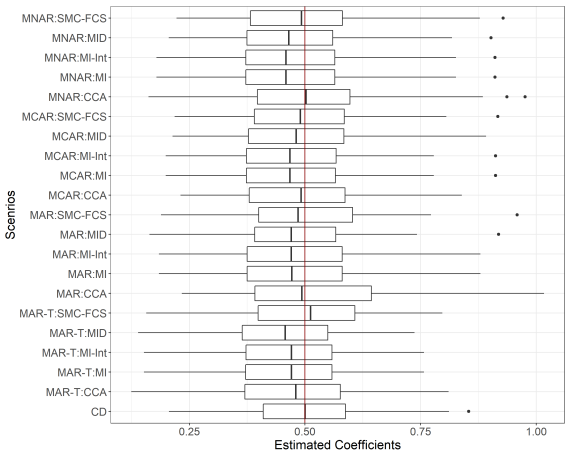


(d) Coefficients of Z_c

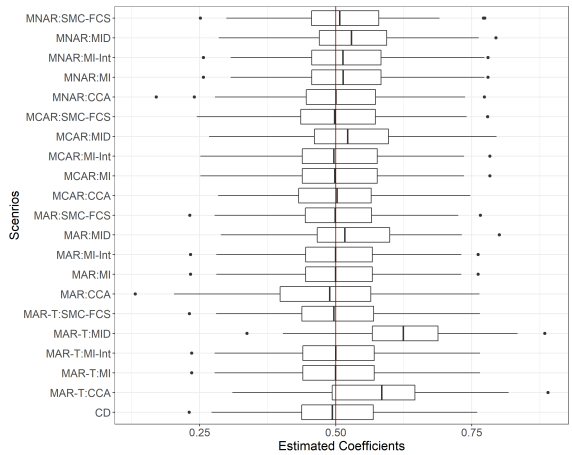
Figure 12: Box plots of Monte Carlo estimated coefficients in different scenarios when stength of association β_1 is equal to 0.05.



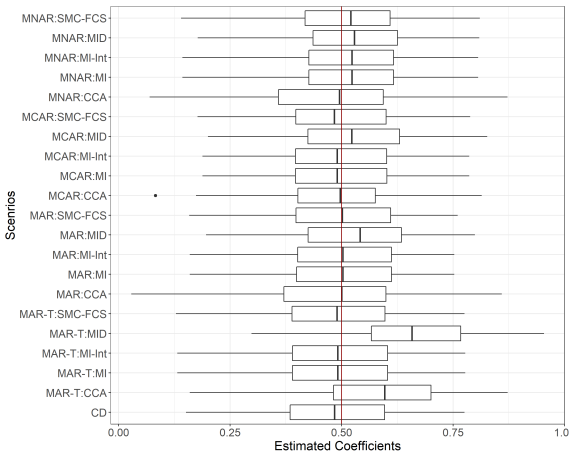
(a) Coefficients of X_b



(b) Coefficients of X_c

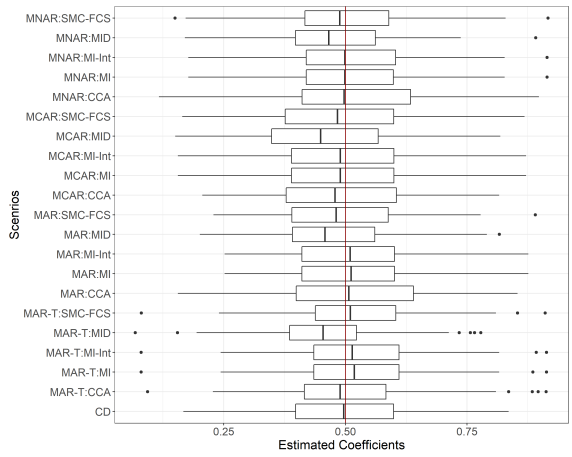


(c) Coefficients of Z_b

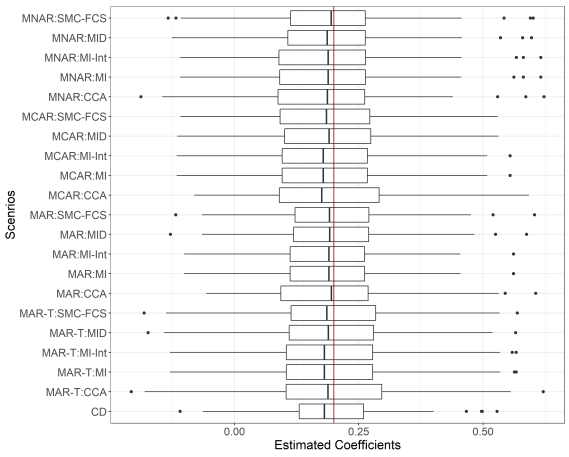


(d) Coefficients of Z_c

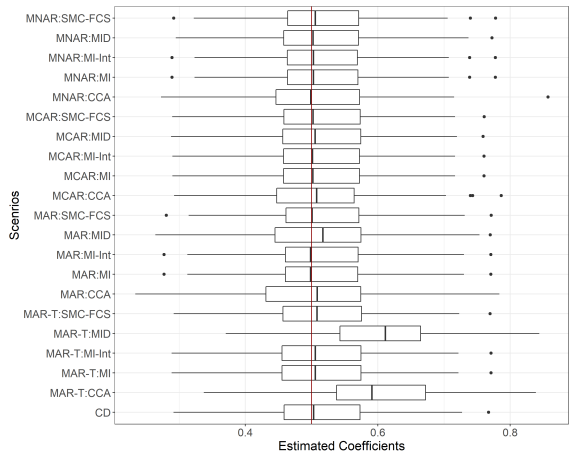
Figure 13: Box plots of Monte Carlo estimated coefficients in different scenarios when strength of association β_1 is equal to 0.5.



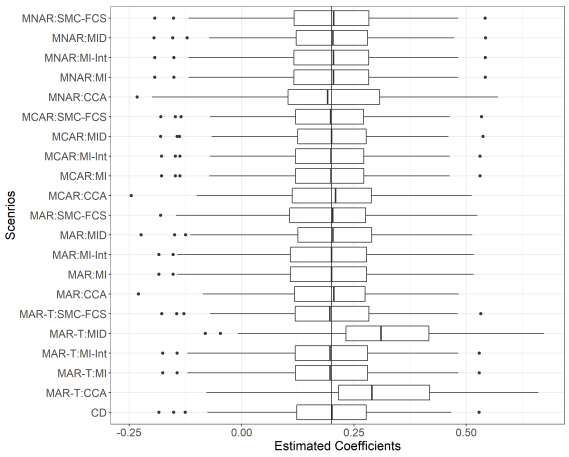
(a) Coefficients of X_b



(b) Coefficients of X_c

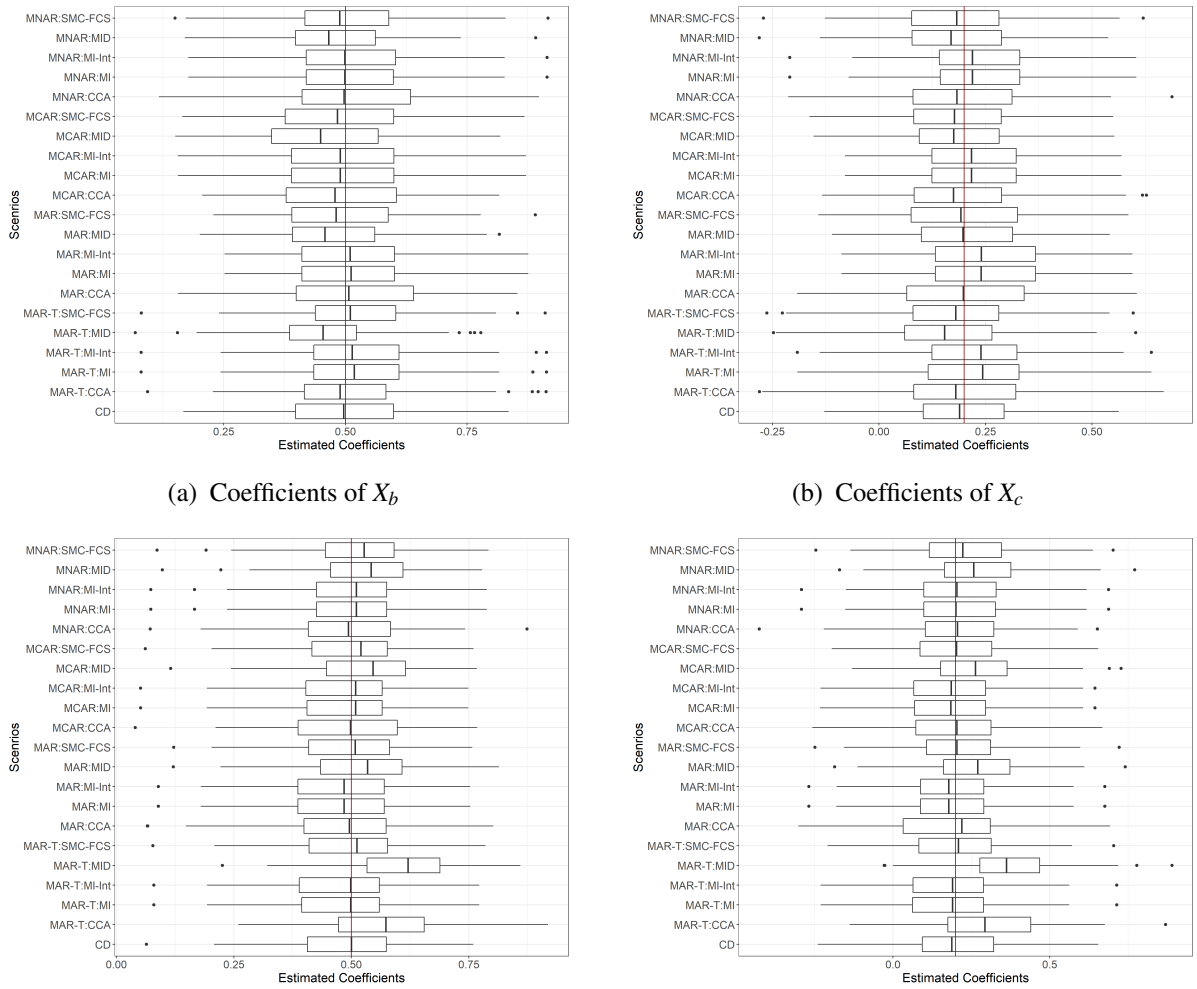


(c) Coefficients of Z_b



(d) Coefficients of Z_c

Figure 14: Box plots of Monte Carlo estimated coefficients in different scenarios when correlation between covariates is 0.01.



(a) Coefficients of X_b

(b) Coefficients of X_c

(c) Coefficients of Z_b

(d) Coefficients of Z_c

Figure 15: Box plots of Monte Carlo estimated coefficients in different scenarios when correlation between covariates is 0.5.

R code

Simulation part

Generating data

```
#############Functions############
#input: dataset: for lapply function, this could be ignored; N is the number
    of observations; kai & lambda: the shape and scale parameter of Weibull
    distribution for survival analysis; no_bin & no_norm: the number of binary
    and continuous variables respectively; pbin: a vector of probabilities for
    the binary variables; normean & norvar: a vector of means/variances for the
     normal variables; betas: the true coefficients of continuous variables
```

```r
#output: data frame with these columns: survival time, status, Z_c, X_c, X_b,Z
    _b
sim_3<-function(dataset,N,kai,lambda,s,no_bin,no_norm,cormat,pbin,normean,
    norvar,betas){
  #covariate part
  covaria<- genpb(n = N,no_bin = no_bin,
                  no_norm = no_norm,inter.mat = cormat,prop_vec_bin = pbin,
                  nor.mean = normean,nor.var = norvar)$data

  covaria_df <- as.data.frame(abs(covaria))
  colnames(covaria_df) <- c("Z_c","X_c","X_b","Z_b")
  #scaled continuous variable
  covaria_df$Z_c <- as.numeric(scale(covaria_df$Z_c,scale = F)/10)
  covaria_df$X_c <- as.numeric(scale(covaria_df$X_c,scale = F)/10)

  u <- runif(N)
  #x1 -- miss x2 -- com
  beta  <- matrix(c(rep(betas,2),0.5,0.5),nrow = 4)
  Y <- (-log(1-u) / (lambda * exp(as.matrix(covaria_df) %*% beta)))^(1/kai)
  C <- rexp(n = N,rate = s)
  YC <- cbind(Y,C)
  time_c <- apply(YC, 1, min)
  status_c <- I(Y <= C) *1

  #artificial censored
  time_c <- ifelse(time_c >= 72,72,time_c)
  status_c <- ifelse(time_c >72,0,status_c)
  data <- as.data.frame(cbind(time_c,status_c))

  covaria_df$X_b <- as.factor(covaria_df$X_b)
  covaria_df$Z_b <- as.factor(covaria_df$Z_b)
  data <- data.frame(time_c= time_c,status_c = status_c)

  data_df <- cbind.data.frame(data,covaria_df)
  data_df$status_c <- as.numeric(data_df$status_c)
  return(data_df)
}
################ Adjusted function according to package PoisBinOrdNor
    ###############
# simulates a multivariate data set that is composed of binary and continuous
    variables with specified marginals and a correlation matrix.
# input: inter.mat: the intermediate correlation matrix obtained from function
     reintermat; others are same with sim_3
# output: covariates data frame

genpb <- function (n, no_bin,no_norm, inter.mat = NULL,
          prop_vec_bin = NULL, nor.mean = NULL, nor.var = NULL)
{
  n2 = no_bin
  n4 = no_norm
  d = n2 +n4
  xx1 = rmvnorm(n, rep(0, d), inter.mat)
  if (n2 != 0) {
```

```
    BB = matrix(0, n, n2)
    for (j in 1:n2) {
      for (i in 1:n) {
        if (1 * xx1[i, j] > qnorm(1 - prop_vec_bin[j]))
          BB[i, j] = 1
        else BB[i, j] = 0
      }
    }
  }
  else BB = NULL
  if (n4 != 0) {
    NN = t(t(xx1[, (n2 + 1):d]) * sqrt(nor.var) +
             nor.mean)
  }
  else NN = NULL
  data = cbind(NN,BB)
  final.corr = cor(data)
  result <- list(n.rows = n, prob.bin = prop_vec_bin,
                 nor.mean = nor.mean, nor.var = nor.var,
                 no.bin = n2, no.norm = n4,
                 data = data)
  return(result)
}
############################
# Calculates and assembles the intermediate correlation matrix entries for the
    multivariate normal data
#input: corr.mat: prespecified correlation matrix for the multivariate data
#output: intermediate correlation matrix

reintermat <- function (no_bin, no_norm, corr_mat = NULL,
          prop_vec_bin = NULL, nor_mean = NULL, nor_var = NULL)
{
  if (no_bin != 0) {
    n2 = no_bin
    p1 = prop_vec_bin
  }
  if (no_norm != 0) {
    n4 = no_norm
    normean = nor_mean
    norvar = nor_var
  }
  d = n2+ n4
  if (n2 == 0 && n4 == 0) {
    stop("Number␣of␣variables␣cannot␣all␣be␣zero!\n")
  }
  #check if there is specification problem
  revalidation(n2,n4, corr_mat, p1,normean, norvar)
  inter.mat = diag(nrow(corr_mat))

    if (n2 != 0 &&  n4 != 0) {
    for (i in 1:n2) {
      for (j in 1:n2) {
        if (i != j) {
```

```
          inter.mat[i, j] = inter.mat[j, i] = corr.nn4bb(p1[i], p1[j], corr_
              mat[i, j])
        }
      }
    }
    inter.mat[(n2 + 1):d, (n2 + 1):d] = corr_mat[(n2+ 1):d, (n2 + 1):d]
    for (i in (n2 + 1):d) {
      for (j in 1: n2) {
        if (i != j) {
          inter.mat[i, j] = inter.mat[j, i] = corr.nn4bn(p1[j], corr_mat[i, j
              ])
        }
      }
    }
  }
  if (!is.positive.definite(inter.mat)) {
    warning("Intermediate␣correlation␣matrix␣is␣not␣positive␣definite.␣Nearest
        ␣positive␣definite␣matrix␣is␣used!")
    inter.mat = as.matrix(nearPD(inter.mat, corr = TRUE,
                                 keepDiag = TRUE)$mat)
    inter.mat = (inter.mat + t(inter.mat))/2
  }
  return(inter.mat)
}

##############################
# computes the lower and upper bounds for all possible pairs that involve
   binary and normal variables.
# output: returns TRUE if no specification problem is encountered
revalidation <- function (no.bin, no.norm, corr.mat = NULL,
          prop.vec.bin = NULL,nor.mean = NULL, nor.var = NULL)
{ n2 = no.bin
  n4 = no.norm
  d = n2 +n4

  is.wholenumber <- function(x, tol = .Machine$double.eps^0.5) abs(x -
                                                                   round(x))
                                                                       < tol
  p = prop.vec.bin
  q = 1 - p
  sigma = corr.mat
  L_sigma = diag(d)
  U_sigma = diag(d)
  u = runif(1e+05, 0, 1)

  if (no.bin > 0) {
    for (i in 1:n2) {
      for (j in 1:n2) {
        if (i != j)
          L_sigma[i, j] = L_sigma[j, i] = max(-sqrt((p[i] * p[j])/(q[i] * q[j
              ])),
                                              -sqrt((q[i] * q[j])/(p[i] *p[j])
                                                  ))
```

```
        if (i != j)
          U_sigma[i, j] = U_sigma[j, i] = min(sqrt((p[i] * q[j])/(q[i] * p[j])
              ),
                                              sqrt((q[i] * p[j])/(p[i] * q[j])
                 ))
      }
    }
  }

  if (no.bin > 0 & no.norm > 0) {
    for (i in (n2+1):d) {
      for (j in 1:n2) {
        if (i != j)
          L_sigma[i, j] = L_sigma[j, i] = -dnorm(qnorm(p[j]))/sqrt(p[j] * q[j
              ])
        if (i != j)
          U_sigma[i, j] = U_sigma[j, i] = dnorm(qnorm(p[j]))/sqrt(p[j] * q[j])
      }
    }
  }
  if (no.norm > 0) {
    for (i in (n2+ 1):d) {
      for (j in (n2 + 1):d) {
        if (i != j)
          L_sigma[i, j] = L_sigma[j, i] = -1
        if (i != j)
          U_sigma[i, j] = U_sigma[j, i] = 1
      }
    }
  }
  valid.state = TRUE
  for (i in 1:d) {
    for (j in 1:d) {
      if (j >= i) {
        if (sigma[i, j] < L_sigma[i, j] | sigma[i, j] >
            U_sigma[i, j]) {
          cat("Range␣violation!␣Corr[", i, ",", j, "]␣must␣be␣between",
              round(L_sigma[i, j], 3), "and", round(U_sigma[i,
                                                            j], 3), "\n")
          valid.state = FALSE
        }
      }
    }
  }
  if (valid.state == TRUE)
    cat("All␣correlations␣are␣in␣feasible␣range!␣\n")
  if (valid.state == FALSE)
    stop("All␣correlations␣must␣be␣in␣feasible␣range!")
  return(TRUE)
}

#################parameters setting#######################
N = 2000#number of patients 200 or 500
```

```
reps<- 1:100 #Monte Carlo replications
max_followup <- 72
imp <- 50 #mice number of datasets
iters <- 2 #mice number of iterations
nbin = 2
nnorm = 2
pbin = c(1/4,1/3)
normean = c(45.87,36.74)
norvar = c(12.56,12.50)
srv_a <- 0.9443
k <- 0.634
MLE_b <- 136.3343  # use surreg function can get 138.1906
surreg_b <- MLE_b^(-k)
lam <- 0.2
s <- 0.08
var_list <- c("Z_c","X_c","X_b1","Z_b1") # comes from the summary of model
pattern_MCAR <- matrix(c(1,1,1,0,1,1,1,1,1,1,0,1,1,1,1,1,0,0,1),nrow = 3,byrow =
    T)
weight_MNAR <- matrix(c(0,0,0,0.2,0,0,0,0,0,0,0,0.2,0,0,0,0,0,0.2,0.2,0),nrow = 3,
   byrow = T)
pattern_MNAR <- matrix(c(1,1,1,0,1,1,1,1,1,1,1,0,1,1,1,1,1,0,0,1),nrow = 3,byrow =
    T)
weight_MAR1 <- matrix(c(0,0,0.2,0,0,0.2,0,0,0.2,0,0,0.2,0,0,0.2,0,0,0.2),nrow
   = 3,byrow = T)
weight_MAR2 <- matrix(c(0.2,0,0.2,0,0,0.2,0.2,0,0.2,0,0,0.2,0.2,0,0.2,0,0,0.2)
   ,nrow = 3,byrow = T)# MAR-T
pattern_MAR <- matrix(c(1,1,1,0,1,1,1,1,1,1,0,1,1,1,1,1,0,0,1),nrow = 3,byrow =
   T)
#----------------correlation matrix--------
corr_1 <- c(0.2,0.2,0.2,0.2,0.2,0.2)# or 0.01,0.5
cor_df_1=diag(4)
cor_df_1[lower.tri(cor_df_1)]=corr_1
ccor_df_1=cor_df_1+t(cor_df_1)
diag(ccor_df_1)<-1
intmat_1 <- reintermat(nbin,nnorm,ccor_df_1,pbin,normean,norvar)

#---------generating 100 datasets---------------
set.seed(2021)
data <- list()
data[["base"]] <- lapply(
  X = reps,
  FUN = sim_3,
  N = N,
  kai = k,
  lambda = surreg_b,
  s = s,
  no_bin = nbin,
  no_norm = nnorm,
  cormat = intmat_1,
  pbin = pbin,
  normean = normean,
  norvar = norvar,
  betas = 0.2 #\beta_1 could be changed to 0.05 or 0.5
```

```
)
```

5.0.0.1 Ampute

```
############functions#######################
#input:X:dataset; missing_prop: missing proportion; pattern_list: A matrix or
    data frame of size patterns by variables where 0 indicates that a variable
    should have missing values and 1 indicates that a variable should remain
    complete; weight_list: A matrix or data frame of size patterns by variables
    . The matrix contains the weights that will be used to calculate the
    weighted sum scores.
#output: data frame with missing values in X_c and Z_c

#ampute missing value ---MCAR
sim_miss <- function(X,missing_prop,pattern_list){
  X_miss <- ampute(X,missing_prop,mech = "MCAR",patterns = pattern_list)$amp
  X_miss$X_b <- as.factor(X_miss$X_b)
  levels(X_miss$X_b ) <- c("0","1")
  X_miss$Z_b <- as.factor(X_miss$Z_b)
  levels(X_miss$Z_b) <-c("0","1")
  return(X_miss)}

#ampute missing value ---MNAR
sim_miss_MNAR <- function(X,missing_prop,pattern_list,weight_list){
  X_miss <- ampute(X,missing_prop,mech = "MNAR",patterns = pattern_list,
     weights = weight_list)$amp
  X_miss$X_b <- as.factor(X_miss$X_b)
  levels(X_miss$X_b ) <- c("0","1")
  X_miss$Z_b <- as.factor(X_miss$Z_b)
  levels(X_miss$Z_b) <-c("0","1")
  return(X_miss)}

#ampute missing value ---MAR & MAR-T
sim_miss_MAR <- function(X,missing_prop,weight_list,pattern_list){
  X$X_b <- as.numeric(X$X_b)
  X$Z_b <- as.numeric(X$Z_b)
  X_miss <- ampute(X,missing_prop,mech = "MAR",weights = weight_list,patterns
     = pattern_list)$amp
  X_miss$X_b <- as.factor(X_miss$X_b)
  levels(X_miss$X_b ) <- c("0","1")
  X_miss$Z_b <- as.factor(X_miss$Z_b)
  levels(X_miss$Z_b) <-c("0","1")
  return(X_miss)}

#missing_prop could change to 0.6
#-------------amupte missing values under MCAR mechanism-------------
missing_df <- list()
missing_df[["MCAR"]] <- lapply( X = data[["base"]],
                                FUN = sim_miss,missing_prop = 0.2,
                                pattern_list = pattern_MCAR)
#-------------amupte missing values under MNAR mechanism-------------
```

```
missing_df[["MNAR"]] <- lapply( X = data[["base"]], FUN = sim_miss_MNAR,
missing_prop = 0.2 ,pattern_list=pattern_MNAR,weight_list=weight_MNAR)
#-------------amupte missing values under MAR mechanism-------------
missing_df[["MAR"]] <- lapply( X = data[["base"]], FUN = sim_miss_MAR,
missing_prop = 0.2,weight_list = weight_MAR1,pattern_list = pattern_MAR )
#-------------amupte missing values under MAR-T mechanism-------------
missing_df[["MAR2"]] <- lapply( X = data[["base"]], FUN = sim_miss_MAR,
missing_prop = 0.2,weight_list = weight_MAR2,pattern_list = pattern_MAR )
```

5.0.0.2 Analysis

```
########function#######
# analysis in complete data
full_fun <- function(data,meth = "breslow"){
  coxph(Surv(time_c, status_c)  ~ Z_c+X_c+X_b+Z_b,
        data = data, method = meth)
}

###model function##-----------------------------------------------------
comp_fun <- function(X){
  index <- !is.na(X$X_b)&!is.na(X$X_c)
  X_comp <- X[index,]
  comp_model <- coxph(Surv(time_c, status_c)  ~ Z_c+X_c+X_b+Z_b,
                      data = X_comp, method = "breslow")
  return(comp_model)
}

missing_indicator <- function(X){
  X_miss <- X
  X_miss$X_b <- as.integer(X_miss$X_b)
  index_1 <- is.na(X_miss$X_b)
  index_2 <- is.na(X_miss$X_c)
  X_miss$X_b[index_1] <- 3
  X_miss$X_b <- as.factor(X_miss$X_b)
  levels(X_miss$X_b) <- c("0","1","2")
  X_miss$X_c[index_2] <- 0
  X_miss$Ad_ind <- rep(0,dim(X_miss)[1])
  X_miss$Ad_ind[index_2] <- 1

  mis_ind_model <- coxph(Surv(time_c, status_c)  ~ Z_c+X_c+X_b+Z_b+Ad_ind,
                         data = X_miss, method = "breslow")
  return(mis_ind_model)
}


mice_fun <- function(data,imp,iters){
  missl <- colnames(data)[sapply(data,anyNA)]
  data$haz_os <- nelsonaalen(data, time_c, status_c)
  pred_mat <- matrix(1,ncol(data),ncol(data),dimnames = list(names(data),names
      (data)))
  diag(pred_mat) <- 0
```

```
  pred_mat[!(rownames(pred_mat) %in% missl),] <- 0
  non_pred <- c("time_c","status_c","Z_c", "Z_b")
  pred_mat[,!(colnames(pred_mat) %in% non_pred)] <- 0

  imputation <- mice(data, maxit = iters, m = imp, seed = 2021, pred = pred_
     mat, print = T)
  mice1_model <- with(imputation,
                      coxph(Surv(time_c, status_c)  ~ Z_c+X_c+X_b+Z_b))
  return(pool(mice1_model))
}

mice_int_fun <- function(data,imp,iters){
  missl <- colnames(data)[sapply(data,anyNA)]
  data$haz_os = nelsonaalen(data, time_c, status_c)

  compl <- c("Z_c","Z_b")
  haz = rep('haz_os', each = length(compl))

  cc = as.data.frame(data[, compl])
  cc[,2] <- as.numeric(cc[,2])
  inter = data[, haz] * cc
  data[, paste0(names(inter), '.int')] = inter
  pred_mat <- matrix(1,ncol(data),ncol(data),dimnames = list(names(data),names
     (data)))
  diag(pred_mat) <- 0
  pred_mat[!(rownames(pred_mat) %in% missl),] <- 0
  non_pred <- c("time_c","status_c","Z_c", "Z_b")
  pred_mat[,!(colnames(pred_mat) %in% non_pred)] <- 0

  imputation <- mice(data, maxit = iters, m = imp, seed = 2021, pred = pred_
     mat, print = T)
  mice2_model <- with(imputation,
                      coxph(Surv(time_c, status_c)  ~ Z_c+X_c+X_b+Z_b))
  return(pool(mice2_model))
}

smcfcs_fun <- function(data,imp,iters){
  outcome <- c("time_c","status_c")
  predictor <- colnames(data)[!(colnames(data) %in% outcome)]
  formu <- stats::reformulate(termlabels = predictor,  response = "Surv(time_c
     ,␣status_c)" )
  smformu <- c(Reduce(paste, deparse(formu)))

  meth_list <- c("","","","norm","logreg","")
  imput <- smcfcs(data, smtype="coxph", smformula= smformu,method = meth_list,
     numit = iters,m=imp)
  coxfun <- function(imp)coxph(Surv(time_c, status_c)  ~ Z_c+X_c+X_b+Z_b,data
     = imp)
  smcfcs_os <- pblapply(imput$impDatasets,FUN = coxfun)
  return (pool(smcfcs_os))
}

#######functions for summarizing the results and preparing for the plots######
```

```r
# the estimating coefficients, CIs from complete data

full_df <- function(model,varible){
  as.data.frame(summary(model)$coef[,1:3],row.names = varible)
} #coef, lower.95, upper.95

#let the coefficients and Cis of five methods be fitted function ggplot
sum_tab <- function(coxsum_df){
  coxsum_df <- as.data.frame(coxsum_df)
  term <- as.character(coxsum_df$term)
  esti <- coxsum_df$estimate
  HR <- exp(coxsum_df$estimate)
  se <- coxsum_df$std
  mat <- as.data.frame(cbind(esti,HR,se))
  rownames(mat) <- term
  return(mat)
}
# summarize the coefficients and CIs of 5 methods
fit_models <- function(data,model,varible){
        #data is for fuction lapply
  if (model == "complete_cases" ){
    fit <- comp_fun(data)
  }else if (model == "missing_indicator"){
    fit <- missing_indicator(data)
  }else if (model == "MICE"){
    fit <- mice_fun(data,50,2)
  }else if (model == "MICE_intern"){
    fit <- mice_int_fun(data,50,2)
  }else {
    fit <- smcfcs_fun(data,50,2)
  }
  if (model == "complete_cases" | model == "missing_indicator"){
    data.frame(      esti = summary(fit)$coef[varible,"coef"],
                     HR = summary(fit)$coef[varible,"exp(coef)"],
                     se = summary(fit)$coef[varible,"se(coef)"],row.names =
                        varible)}
  else{sum_tab(list(summary(fit)))}
}

#--------------analysis with complete data-----------
full_model <- list()
full_model[["base"]] <- lapply(
  X = data[["base"]],
  FUN = full_fun
)

#------------------impute methods-----
#--use MCAR as an example other scenarios are quite same----
MCAR_result <- list()
MCAR_result [["complete_cases"]] <-lapply(
  X = missing_df[["MCAR"]], #change to MAR, MNAR,MAR2
  FUN = fit_models,
  model = "complete_cases",
```

```
  varible = var_list)


MCAR_result [["missing_indicator"]] <-lapply(
  X = missing_df[["MCAR"]],#change to MAR, MNAR,MAR2
  FUN = fit_models,
  model = "missing_indicator",
  varible = var_list)

MCAR_result [["MICE"]] <-lapply(
  X = missing_df[["MCAR"]],#change to MAR, MNAR,MAR2
  FUN = fit_models,
  model = "MICE",
  varible = var_list)

MCAR_result [["MICE_intern"]] <-lapply(
  X = missing_df[["MCAR"]],#change to MAR, MNAR,MAR2
  FUN = fit_models,
  model = "MICE_intern",
  varible = var_list)

MCAR_result [["smcfcs"]] <-lapply(
  X = missing_df[["MCAR"]],#change to MAR, MNAR,MAR2
  FUN = fit_models,
  model = "smcfcs",
  varible = var_list)

#------------------------perpare for calculating bias and RMSE----------
full_HRdf <- lapply(
  X = full_model[["base"]], #change to MAR, MNAR,MAR2
  FUN = full_df,
  varible = var_list)

MCAR_bias <- plot_bias(MCAR_result,100)$df
MCAR_RMSE <- plot_RMSE(MCAR_result,100)$df
#----------------------boxplot-----------------------
###################function###################
#input: all estimated coefficients of specific variable
#output: box plot of coefficients distribution of one variable unber a
    specific scenario

rep_coef <- function(result,varible,rep = 100){
  #varible: 1-Z_c, 2-X_c, 3-X_b, 4-Z_b
  result_list <- numeric(rep)
  for (i in 1:rep) {
    result_list[i] <- result[[i]][,1][varible]
  }
  return(result_list)
}
########use X_b in baseline model under 4 missingness mechianisms and 5
    methods#######
meth_list <- c("complete_cases","missing_indicator","MICE","MICE_intern","
    smcfcs")
```

```
CD_Zb <- rep_coef(full_HRdf,4)
MCAR_Zb <- matrix(unlist(lapply(MCAR_result,rep_coef,varible = 4)),nrow = 100,
    ncol = 5)
MNAR_Zb <- matrix(unlist(lapply(MNAR_result_1,rep_coef,varible = 4)),nrow =
    100,ncol = 5)
MAR_Zb <- matrix(unlist(lapply(MAR_result_1,rep_coef,varible = 4)),nrow = 100,
    ncol = 5)
MAR_T_Zb <- matrix(unlist(lapply(MAR_result_2,rep_coef,varible = 4)),nrow =
    100,ncol = 5)

Zb_res <- as.data.frame(cbind(CD_Zb,MCAR_Zb,MNAR_Zb,MAR_Zb,MAR_T_Zb))
MCAR_label <- c("MCAR:CCA","MCAR:MID","MCAR:MI","MCAR:MI-Int","MCAR:SMC-FCS")
colnames(Zb_res) <- c("CD",MCAR_label,gsub("MCAR","MNAR",MCAR_label),
                      gsub("MCAR","MAR",MCAR_label),gsub("MCAR","MAR-T",MCAR_
                          label))
Zb_res_long <- Zb_res%>%pivot_longer(col = 1:21, names_to = "Scenrios",values_
    to = "Estimated_Coefficients")
Zb_plot <- ggplot(Zb_res_long ,aes(x=Scenrios,y= Estimated_Coefficients)) +
    geom_boxplot() +
  geom_hline(yintercept = 0.5,col="darkred") + coord_flip()+labs(y="Estimated␣
      Coefficients")+
  theme_bw()+
  theme(axis.title = element_text(size=16),axis.text = element_text(size = 14)
      )
ggsave("Zb_missprop.png",plot = Zb_plot,width =  10,height = 8 )

###########functions for bias and RMSE###############
#calculate bias and plot the bias under different scenrios
plot_bias <- function(result_list,rep,beta,full_model = full_HRdf){
  bias_bench <- data.frame(row.names = var_list)
  for (i in 1:rep) {
    bias_bench[,i] <-  full_model[[i]][,1]- c(beta,beta,0.5,0.5)

  }
  bias_bench_val <- apply( bias_bench, 1, mean)

  bias_comp <- data.frame(row.names = var_list)
  for (i in 1:rep) {
    bias_comp[,i] <-  result_list[["complete_cases"]][[i]][["esti"]]-c(beta,
        beta,0.5,0.5)

  }
  bias_comp_val <- apply( bias_comp, 1, mean)
  bias_mis_ind <- data.frame(row.names = var_list)
  for (i in 1:rep) {
    bias_mis_ind [,i] <-   result_list[["missing_indicator"]][[i]][["esti"]]-c
        (beta,beta,0.5,0.5)

  }
  bias_mis_ind_val <- apply(   bias_mis_ind, 1, mean)
  bias_mice <- data.frame(row.names = var_list)
  for (i in 1:rep) {
```

```
    bias_mice[,i] <-   result_list[["MICE"]][[i]][["esti"]]-c(beta,beta
        ,0.5,0.5)

  }
  bias_mice_val <- apply( bias_mice, 1, mean)
  bias_mice_int <- data.frame(row.names = var_list)
  for (i in 1:rep) {
    bias_mice_int[,i] <- result_list[["MICE_intern"]][[i]][["esti"]]-c(beta,
        beta,0.5,0.5)

  }
  bias_mice_int_val <- apply( bias_mice_int, 1, mean)

  bias_smcfcs <- data.frame(row.names = var_list)
  for (i in 1:rep) {
    bias_smcfcs [,i] <-  result_list[["smcfcs"]][[i]][["esti"]]-c(beta,beta
        ,0.5,0.5)

  }
  bias_smcfcs_val <- apply( bias_smcfcs, 1, mean)
  bias <-  as.data.frame(rbind(unlist( bias_bench_val),unlist( bias_comp_val),
      unlist( bias_mis_ind_val),
                               unlist( bias_mice_val),unlist( bias_mice_int_
                                   val),unlist( bias_smcfcs_val)))
  bias$model <- c("CD","CCA","MID","MI","MI-Int","SMC-FCS")


  bias <-  bias %>%pivot_longer(cols = -model,names_to = "term",values_to = "
      bias")
  bias$term <- as.factor(bias$term)
  bias <- as.data.frame( bias[order(bias$term),])
  bias$label <- 1:24
  pp <- ggplot(data =  bias,aes(x=label,y=bias,color = model,shape = term))+
      geom_point() +
    scale_color_manual(values=c("darkred", "blue","green" ,"orange","purple"))
        +
    xlab("Covarite") + ylab("Coefficient␣Bias")+
    geom_hline(yintercept = 0,lty = 2,col = "red")+
    theme(panel.grid =element_blank(),panel.background = element_rect(fill = "
        transparent")) +
    theme(axis.line = element_line(size=0.5, colour = "grey")) +
    scale_x_continuous(breaks=seq(3, 25, 6),labels = c("X_b","X_c","Z_b","Z_c"
        ))

  return(list(pp=pp,df=bias))
}


#calculate RMSE and plot the RMSE under different scenrios
plot_RMSE <- function(result_list,rep = 100,beta, full_model = full_HRdf){
  RMSE_bench <- data.frame(row.names = var_list)
  for (i in 1:rep) {
    RMSE_bench[,i] <- (full_model[[i]][,1]-c(beta,beta,0.5,0.5))^2
```

```
}
RMSE_bench_val <- sqrt(apply(RMSE_bench, 1, mean))

RMSE_comp <- data.frame(row.names = var_list)
for (i in 1:rep) {
  RMSE_comp[,i] <- (result_list[["complete_cases"]][[i]][,1]-c(beta,beta
     ,0.5,0.5))^2

}
RMSE_comp_val <- sqrt(apply(RMSE_comp, 1, mean))
RMSE_mis_ind <- data.frame(row.names = var_list)
for (i in 1:rep) {
  RMSE_mis_ind [,i] <- (result_list[["missing_indicator"]][[i]][,1]-c(beta,
     beta,0.5,0.5))^2

}
RMSE_mis_ind_val <- sqrt(apply(  RMSE_mis_ind, 1, mean))
RMSE_mice <- data.frame(row.names = var_list)
for (i in 1:rep) {
  RMSE_mice[,i] <- (result_list[["MICE"]][[i]][,1]-c(beta,beta,0.5,0.5))^2

}
RMSE_mice_val <- sqrt(apply(RMSE_mice, 1, mean))
RMSE_mice_int <- data.frame(row.names = var_list)
for (i in 1:rep) {
  RMSE_mice_int[,i] <- (result_list[["MICE_intern"]][[i]][,1]-c(beta,beta
     ,0.5,0.5))^2

}
RMSE_mice_int_val <- sqrt(apply(RMSE_mice_int, 1, mean))

RMSE_smcfcs <- data.frame(row.names = var_list)
for (i in 1:rep) {
  RMSE_smcfcs [,i] <- (result_list[["smcfcs"]][[i]][,1]-c(beta,beta,0.5,0.5)
     )^2

}
RMSE_smcfcs_val <- sqrt(apply(RMSE_smcfcs, 1, mean))
RMSE<-as.data.frame(rbind(unlist(RMSE_bench_val),unlist(RMSE_comp_val),
    unlist(RMSE_mis_ind_val),
                              unlist(RMSE_mice_val),unlist(RMSE_mice_int_val)
                                 ,unlist(RMSE_smcfcs_val)))
RMSE$model <- c("CD","CCA","MID","MI","MI-Int","SMC-FCS")

RMSE <- RMSE %>%pivot_longer(cols = -model,names_to = "term",values_to = "
    RMSE")
RMSE$term <- as.factor(RMSE$term)
RMSE <- as.data.frame( RMSE[order(RMSE$term),])
RMSE <- as.data.frame(RMSE)
RMSE$label <- 1:24
pp <- ggplot(data = RMSE,aes(x=label,y=RMSE,color = model,shape = term))+
    geom_point() +
```

```
    xlab("covariate") + ylab("Coefficient␣RMSE")+
    geom_hline(yintercept = 0,lty = 2,col = "red")+
    theme(panel.grid =element_blank(),panel.background = element_rect(fill = "
       transparent")) +
    theme(axis.line = element_line(size=0.5, colour = "grey")) +
    scale_x_continuous(breaks=seq(3, 25, 6),labels = c("X_b","X_c","Z_b","Z_c"
       ))

  return(list(pp=pp,df=RMSE))
}
##############################
```

5.0.1 Application

```
sd <- Hmisc::spss.get("D:/testbook/internship/case/CMLHaplovsMRD_UD_20210713
   NEC.sav", allow = '_', to.data.frame = TRUE, use.value.labels = TRUE,
   datevars = c("datdiag1", "datallo1", "datallo1", "DATCRGR2_allo1", "DPLAT20
   _allo1", "DPLAT50_allo1", "DATRESP_allo1", "DATAGVH_allo1","datcgvhd_allo1"
   ,"datrel_1_allo1","datlast"), max.value.labels = 30)



sd$age_allo1       <- as.numeric(as.character(sd$age_allo1))
sd$AGEDONOR_allo1_1   <- as.numeric(as.character(sd$AGEDONOR_allo1_1 ))
label(sd$age_allo1 ) <- "Age␣at␣allo␣HCT"

#donor type
sd$donrel_ori <- factor(sd$donrel)
sd$donrel <- factor(with(sd, ifelse(as.character(donrel) %in% "Identical␣
   sibling" ,"MRD",ifelse(as.character(donrel) %in% "Haplo","HD",ifelse(as.
   character(donrel) %in% "MUD","MUD","MMUD")))),levels = c("MRD","HD","MUD","
   MMUD"))


# if max_followup needed
max_followup = 72
# time vector
timevect = c(24, 48, 60, 72)
#Exclude patients transplanted with MRD/MUD/MMUD
noverlap <- (sd$cyclophos_prophy_allo1 == "no")|(sd$donrel == "HD")
sd <- sd[noverlap,]

#Exclude patients without donor type
sd <- sd[!is.na(sd$donrel),]
#######---------perpare the data------------------
sd$karnofskcat2_allo1 <- factor(with(sd,ifelse(is.na(KARNOFSK_allo1), NA,
   ifelse(KARNOFSK_allo1%in%c("Normal,␣NED","Normal␣activity␣/␣Minor␣
   restrictions␣in␣strenous␣physical␣activity"),">=90","<90"))),levels = c("
   >=90","<90"))
```

```
sd$cmv_pat_allo1 <- factor(with(sd,ifelse(is.na(cmv_combi_allo1_1),NA,ifelse(
    cmv_combi_allo1_1%in% c("-/-","-/+"),"-","+"))),levels = c("-","+"))
labsource   <- label(sd$source_allo1)

sd$source_allo1 <- droplevels(sd$source_allo1)
label(sd$source_allo1) <- labsource

#source has no missing value,delet PB+BM group here
sd$catsource_allo1 <- factor(with(sd, ifelse(source_allo1 %in% "PB", "PB",
    ifelse(source_allo1 %in% "BM","BM", NA))), levels = c("PB", "BM"))

sd$catyear_allo1 <- factor(sd$YEAR_allo1)
# Status and time variables for overall survival##########artificial censored
sd$srv_s <- ifelse(sd$srv_s_allo1 %in% "dead", 1, 0)
sd$srv_t <- sd$srv_allo1

sd$srv_tc <- ifelse(sd$srv_allo1 > max_followup, max_followup, sd$srv_allo1)
sd$srv_sc <- ifelse(sd$srv_allo1 > max_followup, 0, sd$srv_s)
sd$stagecat3_allo1 <- factor(with(sd,ifelse(as.character(stagecat_allo1) %in%
    "CP1","CP1",
     ifelse(as.character(stagecat_allo1) %in% c("CP2","CP3␣or␣higher","CP␣
        undefined␣nr"),"CP2␣or␣more",
     ifelse(as.character(stagecat_allo1) %in% "AP","AP",
     ifelse(as.character(stagecat_allo1) %in% "other"|is.na(stagecat_allo1), NA
        ,"BC"))))),levels = c("CP1","CP2␣or␣more","AP","BC"))

# Set "Haplo" as the baseline group
sd$donrel <- factor(sd$donrel,levels = c("HD","MRD","MUD","MMUD"))
# complete cases analysis

com_items <- !is.na(sd$source_allo1)&!is.na(sd$ric_allo1)&!is.na(sd$stagecat3_
    allo1)&!is.na(sd$karnofskcat2_allo1)&!is.na(sd$cmv_pat_allo1)&!is.na(sd$age
    _allo1)
de_items <- !com_items
sd$age2_allo1 <- (sd$age_allo1 - 40)/10 #scaled
sd_MA <- droplevels(sd[com_items,c("donrel", "age2_allo1", "stagecat3_allo1",
    "srv_tc", "srv_sc", "ci_tc", "ci_sc", "catsource_allo1", "cmv_pat_allo1", "
    ric_allo1","karnofskcat2_allo1","catyear_allo1")])

coxph.com.os  <- coxph(Surv(srv_tc, srv_sc)  ~ age2_allo1 + donrel +
    karnofskcat2_allo1 + cmv_pat_allo1 + ric_allo1 + stagecat3_allo1 +
    catsource_allo1, data = sd_MA, method = "breslow")
###############check the PH assumption#############
summary(coxph.com.os)
os.com.ph <- cox.zph(coxph.com.os, terms = FALSE)
os.com.ph


# ########################missing indicator
sd_Mind <- sd
sd_Mind$ric_allo1 <- factor(with(sd,ifelse(is.na(ric_allo1),"missing",ifelse(
    as.character(ric_allo1)%in% "standard","standard","reduced"))),levels = c("
    standard","reduced","missing"))
```

```
sd_Mind$cmv_pat_allo1 <- factor(with(sd,ifelse(is.na(cmv_combi_allo1_1),"
    missing",ifelse(cmv_combi_allo1_1%in% c("-/-","-/+"),"-","+"))),levels = c(
    "-","+","missing"))

sd_Mind$stagecat3_allo1 <- factor(with(sd,ifelse(as.character(stagecat_allo1)
    %in% "CP1","CP1",
     ifelse(as.character(stagecat_allo1) %in% c("CP2","CP3 or higher","CP
        undefined nr"),"CP2 or more",
     ifelse(as.character(stagecat_allo1) %in% "AP","AP",
     ifelse(as.character(stagecat_allo1) %in% "other"|is.na(stagecat_allo1), "
        missing","BC"))))),levels = c("CP1","CP2 or more","AP","BC","missing"))

sd_Mind$karnofskcat2_allo1 <- factor(with(sd,ifelse(is.na(KARNOFSK_allo1), "
    missing",ifelse(KARNOFSK_allo1 %in% c("Normal, NED","Normal activity /
    Minor restrictions in strenous physical activity"),">=90","<90"))),levels =
     c(">=90","<90","missing"))

coxph.missind.os  <- coxph(Surv(srv_tc, srv_sc)  ~ age2_allo1 + donrel +
    karnofskcat2_allo1 + cmv_pat_allo1 + ric_allo1 + stagecat3_allo1 +
    catsource_allo1, data = sd_Mind, method = "breslow")

#####MICE#######################'
sd_MAim <- droplevels(sd[,c("donrel", "catyear_allo1", "age2_allo1", "
    stagecat3_allo1", "srv_tc", "srv_sc", "catsource_allo1", "cmv_pat_allo1", "
    ric_allo1","karnofskcat2_allo1")])

missl <- colnames(sd_MAim)[sapply(sd_MAim,anyNA)]
sd_MAim$haz_os = nelsonaalen(sd_MAim, srv_tc, srv_sc)
sd_MAimcat <- sd_MAim
pred <- matrix(1, ncol(sd_MAimcat), ncol(sd_MAimcat), dimnames = list(names(sd
    _MAimcat), names(sd_MAimcat)))
diag(pred) <- 0
pred[!(rownames(pred) %in% missl),] <- 0
non_pred <- c("dornel","age2_allo1","catyear_allo1","srv_tc", "srv_sc")
pred[,!(colnames(pred) %in% non_pred)] <- 0
# number of imputations and iterations
m <- 50
iters <- 5
imputation <- mice(sd_MAimcat, maxit = iters, m = m, seed = 2021, pred = pred,
     print = T)

cox_mice1 <- with(imputation, coxph(Surv(srv_tc, srv_sc) ~ age2_allo1 + donrel
     + karnofskcat2_allo1 + cmv_pat_allo1 + ric_allo1 + stagecat3_allo1 +
    catsource_allo1))

sd_MAimcat2 <- sd_MAim

# ################complete covarites#######################
compl <- c("catyear_allo1","age2_allo1","donrel")
#covarites actually have missing data

haz = rep('haz_os', each = length(compl))
```

```
cc = as.data.frame(sd_MAimcat2[, compl])
cc[,1] <- as.numeric(cc[,1])
cc[,3] <- as.numeric(cc[,3])
inter <- sd_MAimcat2[, haz] * cc
sd_MAimcat2[, paste0(names(inter), '.int')] = inter

pred <- matrix(1, ncol(sd_MAimcat2), ncol(sd_MAimcat2), dimnames = list(names(
    sd_MAimcat2), names(sd_MAimcat2)))
diag(pred) <- 0
pred[!(rownames(pred) %in% missl),] <- 0
non_pred <- c("dornel","srv_tc", "srv_sc", "ci_tc", "ci_sc","age2_allo1","
    catyear_allo1")
pred[,!(colnames(pred) %in% non_pred)] <- 0


# number of imputations and iterations
m <- 50
iters <- 5
imputation2 <- mice(sd_MAimcat2, maxit = iters, m = m, seed = 2021, pred =
    pred, print = T)


cox_mice2 <- with(imputation2, coxph(Surv(srv_tc, srv_sc) ~  age2_allo1 +
    donrel + karnofskcat2_allo1 + cmv_pat_allo1 + ric_allo1 + stagecat3_allo1 +
     catsource_allo1))

final2 <- summary(pool(cox_mice2)
final1 <- summary(pool(cox_mice1))
##############smcics
sd_MAimsm <- droplevels(sd[,c("donrel", "catyear_allo1", "age2_allo1", "
    stagecat3_allo1", "srv_tc", "srv_sc", "ci_tc", "ci_sc", "catsource_allo1",
    "cmv_pat_allo1", "ric_allo1","karnofskcat2_allo1")])

outcomes <- c("srv_tc", "srv_sc")
#outcomes <- missl
#predictors <- c("catyear_allo1")
predictors <- colnames(sd_MAimsm)[!(colnames(sd_MAimsm) %in% outcomes)]


form_os <- stats::reformulate(termlabels = predictors,  response = "Surv(srv_
    tc,␣srv_sc)" )
smform_os <- c(Reduce(paste, deparse(form_os)))
meth <- c("","","","mlogit","","","","","mlogit","logreg","logreg","logreg")
set.seed(2021)
imps <- smcfcs(sd_MAimsm, smtype="coxph", smformula= smform_os,method = meth,
    numit = iters,m=m)
coxfun <- function(imp)coxph(Surv(srv_tc, srv_sc) ~ age2_allo1 + donrel +
    karnofskcat2_allo1 + cmv_pat_allo1 + ric_allo1 + stagecat3_allo1 +
    catsource_allo1,data = imp)
smcfcs_os <- pblapply(imps$impDatasets,FUN = coxfun)
smcfcs_osre <- pool(smcfcs_os)

######################forest plot#################################
```

```
colnames(real_HR)<- c("mean","lower","upper","term","type","term-type")
real_HR1<- real_HR[,c("term","type","mean","lower","upper")]
tab <- matrix(c("Term",as.character(real_HR1$term[1:11])),ncol = 1)
mean_mat <- matrix(real_HR1$mean,11,5)
mean_mat <- rbind(rep(NA,5),mean_mat)
lower_mat <- matrix(real_HR1$lower,11,5)
lower_mat <- rbind(rep(NA,5),lower_mat)
upper_mat <- matrix(real_HR1$upper,11,5)
upper_mat <- rbind(rep(NA,5),upper_mat)

fp <- forestplot(tab,
                 tex_gp = fpTxtGp(ticks = gpar(cex=3),
                                  xlab = gpar(cex=3),
                                  label = gpar(cex=3)),
            legend = c("CCA","MID","MI","MI-Int","SMC-FCS"),
            fn.ci_norm = c(fpDrawNormalCI,fpDrawDiamondCI,fpDrawCircleCI,
               fpDrawDiamondCI,fpDrawCircleCI),
             mean = mean_mat,
              lower = lower_mat,
              upper = upper_mat,
              clip = c(0.5,3),
            lty.ci = c(1,2,1,2,1),
            lwd.ci = 2,
            col = fpColors(box = c("#008B8B", "#FFB90F", "#0000CD","#8A2BE2","
               #8B2323"),lines = c("#008B8B", "#FFB90F", "#0000CD","#8A2BE2","
               #8B2323")),
            vertices = T,
              boxsize = .15,
              ci.vertices = TRUE,
            xticks = c(0.5,1,1.5,2,3),
            xlog = T,
            grid = structure(c(1,2),gp=gpar(lty=3,lwd=2,col="darkgray")),
              xlab = "Hazard␣ratio␣(95%␣CI)")

##################correlation matrix between the variables in real case study
   ##############
#corr
sd_corr <- droplevels(sd[,c( "age_allo1","donrel","karnofskcat2_allo1","cmv_
   pat_allo1", "ric_allo1", "stagecat3_allo1", "catsource_allo1")])
sd_corr <- sapply(sd_corr, as.numeric)

corr_mat <- round(cor(sd_corr,method = "spearman",use = "pairwise.complete.obs
   "),4)
library(corrplot)
corrplot(corr_mat, type = "upper", order = "hclust", tl.col = "black", tl.srt
   = 45)


####################Weibull parameters based on rral case study######
data <- data.frame(cbind(sd$srv_t,sd$srv_s))
data[which(sd$srv_t<=0),] <- 0.0001
colnames(data) <- c("srv_t","srv_s")
death <- data[data$srv_s == 1,]
```

```
x_death <-death$srv_t
death$lnx_div_r <- log(death$srv_t)/length(x_death)

#initial value
shape <- 0.01
for (i in c(1:40)) {
  a <- sum(log(death$srv_t))/length(x_death)
  b <- sum((data$srv_t)^shape)
  c <- sum(((data$srv_t)^shape)*log(data$srv_t))
  d <- sum(((data$srv_t)^shape)*(log(data$srv_t))^2)

  shape <- shape + (a+ (1/shape) - (c/b))/((1/(shape^2))+((b*d) - c^2)/(b^2))
  cat("iter",i,shape,"\n")
}
shape <- max(shape,0.01)
scale <- (sum((data$srv_t^shape)/length(x_death)))^(1/shape)
cat("shape:",shape,"scale:",scale)
```