Alignment between Estimand and Estimator with rbmi

Wenjun Yang, F. Hoffmann-La Roche AG

ABSTRACT

Missing data is commonly seen in clinical trials, and it could also occur if a hypothetical strategy is used to handle certain intercurrent events per ICH E9 (R1). Estimation methods to address the problem presented by missing data should be selected to align with the estimand. rbmi is an R package developed within Roche for implementing imputation methods in continuous multivariate normal longitudinal outcomes under estimand framework.

It supports three imputation approaches, including conventional multiple imputation methods, conditional mean imputation methods, and Bootstrapped multiple imputation methods.

Flexibility of rbmi also allows users to input their own function to leverage the well-established pipeline. rbmi has been tested and validated in multiple cases and shows great strength in its user-friendliness as well as providing a comprehensive solution to multiple imputation for missing data handling.

1. INTRODUCTION

The estimand framework was introduced in ICH E9 (R1) and had been implemented by the different health authorities since 2019 [1]. Intercurrent events and the strategy to handle them were also introduced as one of the five attributes of the estimand, whereas the other four are treatment, population, variable/outcome and population-level summary.

ICH E9 (R1) also distinguishes the concept of missing data from intercurrent events. Some intercurrent events, such as treatment discontinuation, are more likely followed by study discontinuations and missing data. On the other hand, the variables/outcomes, which occur after an intercurrent event and are treated as the hypothetical strategy, would be discarded and as a result the hypothetical variables/outcomes, which would be as if the intercurrent events had not happened, are also unobserved.

To deal with various missing data scenarios, different missing data assumptions are made. For example, missing data due to study withdrawal after the occurrence of a treatment discontinuation due to lack of efficacy may be assumed to be missing not at random because the missingness may be related to unobserved bad treatment outcomes. Another example is that the implementation of the hypothetical strategy often involves the assumption of missing at random whereas the missingness is not related to unobserved outcomes/variables.

When outcomes are multivariate normal, reference based imputation methods, either multiple imputation [2] or conditional mean imputation [3], have been proposed to impute missing variables/outcomes depending on missing scenarios.

rbmi is an open-source package for the statistical programming language R. Developed by Roche, rbmi is an implementation of reference based multiple imputation and conditional mean imputation, which allows both fast implementation and flexibility with its powerful pipeline and user-friendly APIs.

The aim of this paper is to introduce the implementation of rbmi under the framework of estimand. The paper is organized as follows. In Section 2, the required data structures will be introduced and In Section 3, the statistical methods implemented in rbmi will be briefly described. In Section 4, the workflow of using rbmi for analysis is presented. In Section 5, a code example will be provided to show the audience how to quickly start using rbmi. This paper will end with a conclusion.

2. DATA STRUCTURE

In this section, we present the required data to prepare before performing the analysis using rbmi and highlight their structures.

data(Draws): A longitudinal data.frame containing target outcome variable and covariates to be concluded in the analysis.

USUBJID	SEX	TRT01A	BASE	VISIT	OUTCOME	…
10001	F	DRUG A	10	WEEK 2	115	…
…	..	…	…	…	…	…
10002	M	PLACEBO	12	WEEK 2	80	…
…	…	…	…			…

Table 1

Note rbmi expects the covariates to be present for each patient. If not, then a certain fill-in method needs to be performed on the covariate with careful consideration. For example, if estimating treatment effects by region, and the covariates also tend to have different levels across regions, then region might be necessary to consider when filling in covariates.
rbmi also expects the data to be complete, that is to say even though some visits are not observed due to ICE events, but the records should be present in data with the outcome variable as NA. A built-in LOCF function could be used to facilitate this process.

data_ice(Draws): A data.frame which specifies the first visit affected by an intercurrent event (ICE) and the imputation strategy for handling missing outcome data after the ICE per affected patient. For example, if the analysis is Primary analysis, then certain ICE records can be filtered from ADICE(see example below) based on PARCAT1 and sorted by ascending order of time. dat_ice then can be created through taking the first ICE event of each patient and selecting the necessary variables.

An example of dat_ice:

USUBJID	VISIT	STRATEGY
10001	WEEK 8	JR
10002	WEEK 12	JR
10004	WEEK 6	MAR
…	…	…

Table 2

ADICE: a BDS structure analysis dataset specifies the ICEs for each patient. One record per patient and one ICE.

An example of ADICE:

USUBJID	…	PARCAT1	PARCAT2	PARAM	PARAMCD	AVALC	AVISIT	ADT	STRATEGY
10001	…	Primary	SDCR	EOT AE	EOTAE	Y	WEEK 12	10MAR2023	JR
10002	…	Sensitivity	NSDCR	>=2 dose missed due to COVID	COVM2	Y	WEEK 10	20APR2023	CIR
10002	…	Primary	SDCR	EOT Lack of Efficacy	EOTLE	Y	WEEK 12	30MAY2023	JR
…	…			…	…	…	…	…	

Table 3

3. STATISTICAL METHODS IMPLEMENTED

Both multiple imputation (conventional Bayesian[5] and bootstrapped) and conditional mean imputation[3] were implemented in rbmi. Only single imputation was needed in conditional mean imputation, while repetitions were needed in multiple imputation as directly seen in its name. Other than the number of imputation repetitions, four basic steps were involved in each repetition, which are 1) base imputation model fitting step; 2) imputation step; 3) analysis step; 4) pooling step for inference.

Base Imputation Model Fitting Step

In the base imputation model step of the conventional Bayesian multiple imputation method, *M* samples of the estimated parameters (regression coefficients and covariance matrices) will be drawn from a Bayesian posterior distribution of a multivariate normal mixed model for repeated measures fitted to the analysis dataset, where M is the number of multiple imputation.

In the bootstrap multiple imputation, conventional restricted maximum-likelihood parameter estimation (RMLE) of MMRM will be applied to B nonparametric bootstrap samples from the analysis dataset, where B is the number of bootstraps, while a conventional RMLE of MMRM will be fitted to the analysis dataset in conditional mean imputation.

Imputation Step

For each sample of the estimated parameters of the conventional Bayesian multiple imputation method, the joint distribution of all longitudinal outcomes of each subject can be described through the parameter values and the defined imputation strategy, regardless if the outcomes are observed or missing. The distribution of the missing outcomes conditional on the observed outcomes also could be constructed, from which a single sample could be drawn to impute the missing outcomes leading to a complete imputed dataset. The available imputation strategy can be seen in Table 5. Similar steps are also applied in bootstrap multiple imputation and conditional mean imputation, except that D samples will be drawn from the conditional distribution of the missing outcomes in the bootstrap multiple imputation and the conditional mean of the missing outcomes will be used to imputed the missing outcomes in the conditional mean imputation method.

Analysis Step

The conventional Bayesian multiple imputation method will provide M complete imputed dataset from, the bootstrap multiple imputation will provide B×D, and the conditional mean imputation will provide one. Each complete imputed dataset in the conventional Bayesian multiple imputation method will be analyzed through a statistical model (e.g. ANCOVA), resulting in a point estimate and a standard error of the treatment effect.

Pooling Step for Inference

Lastly, the M treatment effect estimates, standard errors, and degrees of freedom will be pooled following the rules by Barnard and Rubin[6] to obtain the final pooled treatment effect estimator, standard error, and degrees of freedom in the conventional Bayesian multiple imputation method. The B×D treatment effect estimates can be pooled as described in von Hippel and Bartlett (2021)[7] to obtain the final pooled treatment effect estimate, standard error, and degrees of freedom Jackknife and the bootstrap method could be used to get the standard error of the estimate from conditional mean imputation.

4. Package Workflow

The pipeline of rbmi can break down into 4 parts: Draws, Impute, Analyze and Pool, which correspond to the basic four steps mentioned in section 3.

- **Draws:**

The draws function corresponds to the base imputation model fitting step. It fits the imputation models and stores the corresponding parameter estimates.There are several built-in imputation methods and strategies, while the end user can also add their own.

Available imputation methods include:

Bayesian multiple imputation	method_bayes()
Approximate Bayesian multiple imputation	method_approxbayes()
Conditional mean imputation (bootstrap)	method_condmean(type = "bootstrap")
Conditional mean imputation (jackknife)	method_condmean(type = "jackknife")
Bootstrapped multiple imputation	method = method_bmlmi()

Table 4

Available imputation strategies include:

Missing At Random	MAR
Jump to Reference	JR
Copy Reference	CR
Copy Increments from Reference	CIR
Last Mean Carried Forward	LMCF

Table 5

```
drawObj = draws(data,
            data_ice,
            vars = set_vars(
                  outcome, visit, subjid, group, covariates),
            method = method_condmean(type = "bootstrap"), ...)
```

- **Impute:**
  The impute function implements the imputation step. It uses the parameters from the imputation model to generate the imputed datasets

```
imputeObj =  impute(drawObj,
                    references = c("DRUG" = "PLACEBO", "PLACEBO" = "PLACEBO"),
                    strategies = getStrategies(),
                    ...)
```

- **Analyze:**
  The analyze function aligns with the analysis step, running the built-in model or your own(through fun variable) on each imputed dataset

```
analysisObj = analyse(imputeObj,
```

```
                    fun = ancova,
                    vars = set_vars(
                         outcome, visit, subjid, group, covariates),
                    ...)
```

- **Pool:**
  The pool function corresponds to the pooling step for inference. It summarizes the analysis results across multiple imputed datasets to provide an overall statistics with a standard error, confidence intervals and a p-value for the hypothesis test of the null hypothesis that the effect is equal to 0. Note that we choose the pooling method automatically based on the imputation method used previously.

```
poolObj = pool(analysisObj,
               conf.level = 0.95,
               alternative = c("two.sided", "less", "greater"),
               type = c("percentile", "normal"))
```

5. CODE EXAMPLE:

In this following example, we will perform reference-based multiple imputation and ANCOVA on the public dataset of antidepressant data. We include a region treatment interaction factor to highlight the flexibility of rbmi. Conditional mean imputation method is used for illustrative purposes.

1) Overview of Data and Model

- **Data:**
  There are 172 patients in total and after random assignment of region, 45 asian patients are on active arm and 41 on placebom, while for non-asian the numbers are 39 and 47 respectively.

- **Fraction of missing data:**
  0% at Visit 4, 8% at Visit 5, 13% at Visit 6, 25% at Visit 7.

- **Model**:
  Reference-based multiple imputation will be performed using conditional mean imputation method with jackknife resampling, in this case, we will have 173 imputed datasets. We assume that the imputation strategy after the ICE is Jump To Reference (JR). The imputation model will include region-treatment-visit interaction. Ancova model will be performed on the imputed datasets, including region-treatment interaction.

2) Load Data

```
library(rbmi)
library(dplyr)
library(assertthat)

data('antidepressant_data')
dat <- antidepressant_data

# 172 patients in total
# Variables available: PATIENT, HAMATOTL, PGIIMP, RELDAYS, VISIT, THERAPY, GENDER,
POOLINV, BASVAL, HAMDTL17, CHANGE
```

3) Preprocessing

```
# Assign region information randomly
{set.seed(777); asia_pt <- sample(dat$PATIENT%>%as.vector()%>%unique(),86)}

dat <- dat%>%
  mutate(REGION = ifelse(PATIENT %in% asia_pt, 'ASIA', 'NASIA'))

#             REGION
# THERAPY   ASIA NASIA
# DRUG      45    39
# PLACEBO   41    47
```

4) Impute Missing Covariates and Complete the Datasets

- rbmi expects the covariates to be present for each patient. We don't have this issue in this test dataset. But it's always nice to check and consider what data to use to fill in the missing covariates.
- rbmi expects the dataset to be complete, meaning the missing data should also have records in the datasets with target outcome equaling NA. Thus here we use rbmi built-in LOCF function to complete the datasets.

```
dat <- expand_locf(dat,
                   PATIENT = levels(dat$PATIENT),
                   VISIT = levels(dat$VISIT),
                   vars = c('BASVAL', 'THERAPY', 'REGION'),
                   group = c('PATIENT'),
                   order = c('PATIENT', 'VISIT'))%>%
  select(PATIENT, VISIT, BASVAL, THERAPY, REGION, CHANGE)
```

5) Prepare ICE datasets

```
dat_ice <- dat %>%
    arrange(PATIENT, VISIT) %>%
    filter(is.na(CHANGE)) %>%
    group_by(PATIENT) %>%
    filter(row_number()==1)%>%
    ungroup() %>%
    select(PATIENT, VISIT) %>%
    mutate(strategy = "JR")
```

6) Draws

```
vars <- set_vars(
    outcome = "CHANGE",
    visit = "VISIT",
    subjid = "PATIENT",
    group = "THERAPY",
    covariates = c("BASVAL*VISIT", "THERAPY*VISIT", "REGION*THERAPY*VISIT")
)

method <- method_condmean(covariance = "us",
           threshold = 0.01,
           same_cov = TRUE,
           REML = TRUE,
           n_samples = NULL,
           type = "jackknife")

set.seed(2023)
drawObj <- draws(
    data = dat,
    data_ice = dat_ice,
    vars = vars,
    method = method,
    quiet = TRUE
)
drawObj
```

7) Impute

```
imputeObj <- impute(
    drawObj,
    references = c("DRUG" = "PLACEBO", "PLACEBO" = "PLACEBO")
)
```

8) Define Customized Analyse Function

Here we define a customized function to perform ANCOVA with region-treatment interaction and capture the difference between treatment groups in each of the regions.

```
ancova_region = function (data, vars, visits = NULL, weights = c("proportional", "equal"))
{
  outcome <- vars[["outcome"]]
  group <- vars[["group"]]
  covariates <- vars[["covariates"]]
  visit <- vars[["visit"]]
  weights <- match.arg(weights)
  if (is.null(visits)) {
    visits <- unique(data[[visit]])
  }
  res <- lapply(visits, function(x) {
    data2 <- data[data[[visit]] == x, ]
```

```
    res <- ancova_single_region(data2, outcome, group, covariates,
                                weights)
    names(res) <- paste0(names(res), "_", x)
    return(res)
  })
  return(unlist(res, recursive = FALSE))
}

ancova_single_region = function (data, outcome, group, covariates, weights =
c("proportional", "equal"))
{
  weights <- match.arg(weights)
  data[[group]] <- as.numeric(data[[group]]) - 1
  data2 <- data[, c(rbmi:::extract_covariates(covariates), outcome, group)]
  frm <- rbmi:::as_simple_formula(outcome, c(group, covariates))
  mod <- lm(formula = frm, data = data2)
  args <- list(model = mod, .weights = weights)

  int_nasia = paste0(group,":REGIONNASIA")

  # region: Asia Non-Asia
  args[[group]] <- 0
  args[["REGION"]] <- "ASIA"
  lsm0asia <- do.call(rbmi:::lsmeans, args)
  args[["REGION"]] <- "NASIA"
  lsm0nasia <- do.call(rbmi:::lsmeans, args)

  args[[group]] <- 1
  args[["REGION"]] <- "ASIA"
  lsm1asia <- do.call(rbmi:::lsmeans, args)
  args[["REGION"]] <- "NASIA"
  lsm1nasia <- do.call(rbmi:::lsmeans, args)

  x <- list(trt_asia = list(est = coef(mod)[[group]],
                            se = sqrt(vcov(mod)[group, group]),
                            df = df.residual(mod)),
            lsm_ref_asia = lsm0asia,
            lsm_alt_asia = lsm1asia,
            trt_nasia = list(est = coef(mod)[[group]] + coef(mod)[[int_nasia]],
                            se = sqrt(vcov(mod)[group, group] + 2*vcov(mod)[int_nasia,
group] + vcov(mod)[int_nasia, int_nasia] ),
                            df = df.residual(mod)),
            lsm_ref_nasia = lsm0nasia,
            lsm_alt_nasia = lsm1nasia
  )
  return(x)
}
```

9) Analyse: Fit ANCOVA Model

```
anaObj <- analyse(
    imputeObj,
```

```
    ancova_region,
    vars = set_vars(
        subjid = "PATIENT",
        outcome = "CHANGE",
        visit = "VISIT",
        group = "THERAPY",
        covariates = c("BASVAL", "REGION*THERAPY")
    )
)
anaObj$results
```

10) Pool

```
poolObj <- pool(
    anaObj,
    conf.level = 0.95,
    alternative = "two.sided"
)
poolObj
```

11) Report

```
=========================================================
      parameter       est     se     lci     uci      pval
---------------------------------------------------------
          …
     trt_asia_7     2.397   1.35   -0.249  5.043   0.076
   lsm_ref_asia_7   -7.533  0.926  -9.347  -5.719  <0.001
   lsm_alt_asia_7   -5.136  1.192  -7.473   -2.8   <0.001
     trt_nasia_7    1.791   1.141  -0.444  4.027   0.116
   lsm_ref_nasia_7  -6.406  1.021  -8.406  -4.406  <0.001
   lsm_alt_nasia_7  -4.615  1.014  -6.602  -2.628  <0.001
---------------------------------------------------------
```

The report gives an estimated difference of 1.791 (95% CI -0.444 to 4.027) between the two treatment groups at the visit 7 in the Non-Asia region with an associated p-value of 0.116. While in the Asia region, the estimated difference is 2.397 (95% CI -0.249 to 5.043) between the two treatment groups with a p-value of 0.076.

CONCLUSION

rbmi is an R package developed by Roche dedicated to help with the problem that the estimation methods to address the problem presented by missing data should be aligned with the estimand, from the programming side. rbmi can help you perform missing data analysis through a neat pipeline with flexibility to tailor to your own scientific question. A code example is given to help better understand the workflow and the highlights of the package.

REFERENCE

[1]. ICH E9 working group. 2019. “ICH E9 (R1): Addendum on estimands and sensitivity analysis in clinical trials to the guideline on statistical principles for clinical trials.” International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use.
[2]. Cro, S., Reference based multiple imputation; for sensitivity analysis of clinical trials with missing data, https://www.stata.com/meeting/uk16/slides/cro_uk16.pdf
[3]. Wolbers, Marcel, Alessandro Noci, Paul Delmar, Craig Gower-Page, Sean Yiu, and Jonathan W. Bartlett. 2022. “Standard and Reference-Based Conditional Mean Imputation.” Pharmaceutical Statistics. https://doi.org/10.1002/pst.2234.
2019. https://database.ich.org/sites/default/files/E9-R1_Step4_Guideline_2019_1203.pdf.
[4]. Gower-Page C, Noci A (2022). rbmi: Reference Based Multiple Imputation. https://insightsengineering.github.io/rbmi/, https://github.com/insightsengineering/rbmi.
[5]. Carpenter, James R, James H Roger, and Michael G Kenward. 2013. “Analysis of Longitudinal Trials with Protocol Deviation: A Framework for Relevant, Accessible Assumptions, and Inference via Multiple Imputation.” Journal of Biopharmaceutical Statistics 23 (6): 1352–71.
[6]. Barnard, John, and Donald B Rubin. 1999. “Miscellanea. Small-Sample Degrees of Freedom with Multiple Imputation.” Biometrika 86 (4): 948–55.
[7]. Von Hippel, Paul T, and Jonathan W Bartlett. 2021. “Maximum Likelihood Multiple Imputation: Faster Imputations and Consistent Standard Errors Without Posterior Draws.” Statistical Science 36 (3): 400–420.

CONTACT INFORMATION:

Your comments and questions are valued and encouraged. Contact the author at:

Wenjun Yang
Wenjun.yang.wy1@roche.com