missingHE: Health Economic Evaluations with Missing Data

Andrea Gabrio
University College London
Department of Statistical Science
Gower Street, London, WC1E 6BT (UK)
E-mail: ucakgab@ucl.ac.uk

Abstract. This is my abstract.

1 Introduction

Health economic evaluations based on individual-patient data are generally characterised by a significant proportion of missing values in the outcome variables. If these unobserved values are not appropriately handled through suitable methods, they may bias results and possibly mislead conclusions. This is particularly important for the modelling of data in Cost-Effectiveness Analyses (CEAs), which typically show very complex dependence structures between measures of clinical benefit, e.g. Quality-Adjusted Life Years (QALYs), and the costs associated with the specific medical intervention considered.

In the *statistical* literature, people deal with missing data using different types of methods, such as Complete Case Analysis, Single Imputation or Multiple Imputation. These are typically based on the assumption that all the information required to obtain valid inferences is embedded in the observed data. Such an assumption, however, is not likely to hold in many practical cases. Thus, in all but the simplest and least realistic examples, missing data analyses need to explicitly incorporate some extra information from sources other than the observed data. This amounts to using untestable assumptions, obtained from the specific framework analysed or via expert opinions, with the purpose to inform and guide the analysis.

The failure to correctly specify the uncertainty related to missingness may have important consequences in terms of CEA decision output. This is extremely relevant from the perspective of bodies responsible for informing the provision of health care, such as the National Institute for Health and Care Excellence (NICE) in the UK. Indeed, such analyses may incorrectly suggest that a new intervention is cost-effective with respect to some comparators. This, in turn, could lead to a reimbursement of the new intervention by the public health care provider (e.g. NHS) that would not otherwise have occurred under more cautious considerations of missing data assumptions. As a result, the final resource allocation decision could be significantly affected.

Bayesian methods are well-suited to addressing decision-making problems, such as that in economic evaluations (Briggs et al., 2006; Briggs and Gray, 1999; Baio, 2013). By taking a probabilistic approach, based on decision rules and available information, they can explicitly account for all forms of uncertainty in the decision process and obtain an “optimal” decision output. The Bayesian approach naturally allows the formalisation and exploration of plausible missingness assumptions to assess the robustness of the results to a range of alternatives. This is possible because, under the Bayesian paradigm, missing data can be thought of as parameters, i.e. unknown variables, whose uncertainty can be addressed via suitably-defined prior distributions. Within this setting we can use a *principled* approach to missingness that embeds critical thinking about missing data assumptions and handles complex dependence structures between variables.

With this in mind, the objective of this work is to develop a suite of functions and tools for the freely available statistical software R, specifically designed to allow the incorporation of external information to explore alternative plausible missingness assumption scenarios.

2 The R package of missingHE

MissingHE is a package designed to aid in the process of economic evaluations in Health Economics in the presence of missing data in the outcome variables. The Bayesian approach is used to assess the uncertainty induced by the missing data and underpinning decision-making problems, from the perspective of decision makers. In fact, **missingHE** can be considered a wrapper for some other R packages. The first two, **R2OpenBUGS** (Gelman et al., 2017) and **R2jags** (YS. and Yajima, 2015), are programs for simulation from Bayesian hierarchical models using Markov chain Monte Carlo (MCMC) methods that are based on the `BUGS` modelling language. The third, **BCEA** (Baio et al., 2016), is used to produce an economic evaluation output from the posterior inference generated via the software `OpenBUGS` or `JAGS`. The package also relies on other packages such as **ggplot2** (Wickham and Chang, 2016), **gridExtra** (Auguie and Antonov, 2016), **ggthemes** (Arnold et al., 2017), **mcmcplots** (Curtis et al., 2015) and **ggmcmc** (Marin, 2016), mainly for graphics purposes.

2.1 Economic Evaluations: A Bayesian approach

With respect to the context analysed, a Bayesian approach has several convenient aspects. For example, health economic evaluations are typically based on complex models, often made by several (correlated) modules, which may be informed by different and diverse sources of evidence. Thus, a Bayesian approach can be beneficial to propagate the uncertainty underlying all model parameters in a principled way. This is also particularly relevant in terms of Probabilistic Sensitivity Analysis (PSA), e.g. the practice of assessing the impact of parameters uncertainty on the decision-making process. PSA is usually based on a simulation approach to characterise the uncertainty of the model parameters, a fundamentally Bayesian operation. On the other hand, Bayesian models require the specification of suitable prior distributions that are consistent with the information available, and can be computationally intensive. However, given the advantages Bayesian models can offer in standard modelling settings we build the whole economic modelling under a Bayesian framework and take full advantage of the flexibility provided by MCMC estimation (Brooks et al., 2011).

2.2 Modellig Missing Data in Economic Evaluations

Missing data do not only impact analyses because of a reduced sample size, possibly affecting efficiency of the estimates, but they also prevent the use of standard complete-data methods. Nonrespondents can be systematically different from respondents in terms of characteristics or outcomes. This could lead to biases in the results that are difficult to eliminate, possibly leading to the (often implicit) formulation of unrealistic missingness assumptions. This is particularly true when, as often happens in clinical studies, reasons for missingness are ignored or not properly recorded. As a result, the statistical and economic analysis will be impaired because assumptions about unobserved data will be forced by the lack of available information about the missing values.

Finally, it is important to stress that, in the presence of missingness, any analysis method implies some assumptions about the missing values, which cannot be ultimately tested from the data at hand. As it has been pointed out (Molenberghs et al., 2015), a convenient way for broadly identifying the severity of the missingness problem is whether reasons for nonresponse are related or not to the outcome of interest. If there is no relation, then missingness should not hugely complicate the analysis, while if such a relationship exists it is plausible for nonrespondents to be systematically different from respondents. This may lead to biases that can distort the results and, within CEAs, lead to the approval of a treatment option that in fact is not cost-effective.

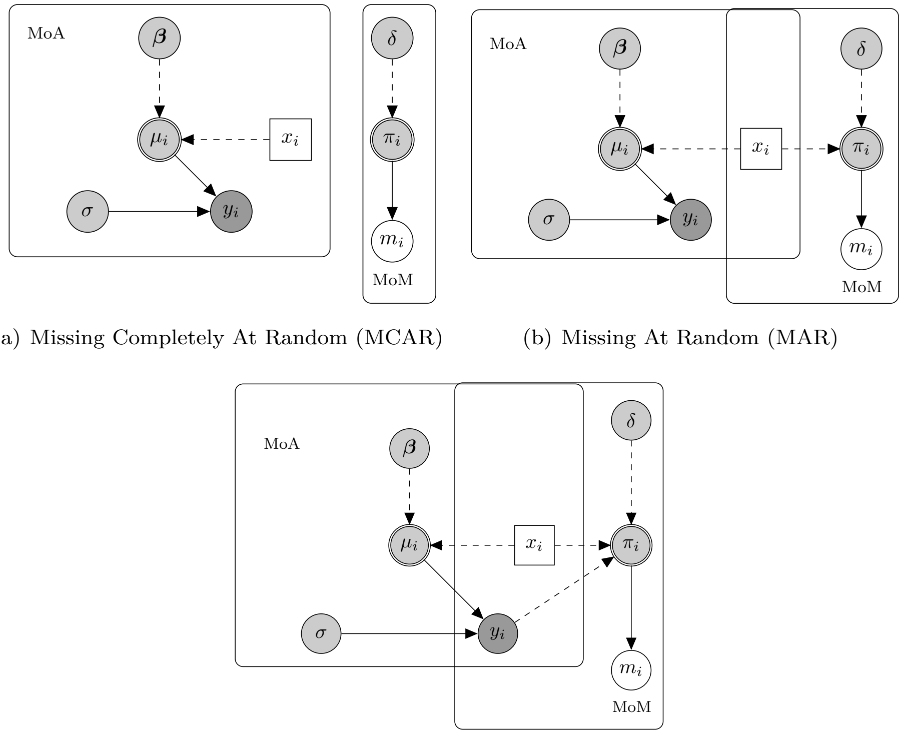
2.3 Missing Data Mechanism

When analysing partially observed data, it is essential to investigate the possible reasons behind missingness. This formally translates into an *assumed* missing data mechanism (Rubin, 1987) that is linked to the data generating process, a key concept to address missingness in a "principled" way. We specifically refer to "principled" methods for missing data as those based on a well-defined statistical model for the complete data, and explicit assumptions about the missing value mechanism.

We consider a sample of $i = 1, \dots, n$ individuals for whom the relevant outcome is indicated as y_i, which is unobserved for some individuals. Typically, trial data also include a set of J covariates $\boldsymbol{x}_i = (x_{1i}, \dots, x_{Ji})$, e.g. sex, age or co-morbidities. While in general these may be partially or fully observed, in this section we consider only the latter case. In addition, we define a missingness indicator m_i taking value 1 if the $i-$th subject is associated with missing outcome and 0 otherwise.

This setting can be modelled using two sub-models, or "modules". The first module is the missing data mechanism, denoted as *Model of Missingness* (MoM). It describes a probability distribution for m_i, as a function of some unobserved parameters π_i and δ, defining the probability of missingness in the outcome variable y_i. The second module is the data generating process of the outcome variable, denoted as *Model of Analysis* (MoA). This contains the main parameters of interest (e.g. the population average costs and benefits) and describes a probability model for the outcome y_i. As a general example, we can think of a simple regression model where $y_i \sim \text{N}(\mu_i, \sigma)$, and $\mu_i = \beta_0 + \beta_1 x_i$. In this case, the parameters of the MoA are the regression coefficients $\boldsymbol{\beta} = (\beta_0, \beta_1)$ representing respectively the intercept and the slope, and the individual standard deviation σ.

The most accepted classification of missingness mechanisms is given by (Rubin, 1987) and is based on three classes, according to how the missingness probability in the MoM is modelled. A simple graphical representation of the three classes is provided in Figure 1. Variables and parameters are denoted by nodes of different shapes and colours according to their nature. Parameters (β_0, β_1, σ, δ) are represented through grey circles. "Logical" quantities such as μ_i and π_i, which are defined as a function of other parameters, are denoted by a double circle notation. Fully observed variables (m_i) are represented with a white circle while partially observed variables (y_i) are denoted by a darker grey circle. Finally, we show covariates (x_i) as white squares to indicate that they are fully observed and not modelled. Rounded rectangles are used to show the extent of the two modules in terms of variables/parameters included. Arrows show the relationships between the nodes, with dashed and solid lines indicating logical functions and stochastic dependence, respectively.



(a) Missing Completely At Random (MCAR)

(b) Missing At Random (MAR)

(c) Missing Not At Random (MNAR)

Figure 1: Graphical representation of Rubin's missing data mechanism classes, namely (a) MCAR, (b) MAR and (c) MNAR. Variables and parameters are represented through nodes of different shapes and colours. Parameters are indicated by grey circles with logical parameters defined by double circles, while predictor variables are assumed fixed and drawn as white squares. Fully observed variables are denoted by white circles, partially observed variables by darker grey circles. Nodes are related to each other through dashed and solid arrows which respectively represent logical functions and stochastic dependence. MoA=Model of Analysis, MoM=Model of Missingness.

The missing data mechanism specifies a probability model for the distribution of m_i conditional on all other variables, broadly distinguished, according to Rubin's classification, into three classes.

Figure 1 (a) illustrates the class of 'Missing Completely At Random' (MCAR), in which the probability of missingness is fully independent of any other partially or fully observed variable. Consequently, in Figure 1 (a) MoA and MoM are not connected and π_i does not depend on any quantity in the MoA. This amounts to assuming that there is no systematic difference between partially and fully observed individuals in terms of the outcome y_i. In other words, in this case we would be assuming that observed cases are a representative sample of the full sample.

Figure 1 (b) shows a case of 'Missing At Random' (MAR), in which the missingness probability may depend on a fully observed variable. As a result, MoA and MoM are connected by means of the predictor variable affecting both the mechanisms generating y_i and m_i. Because of this relationship, the partially observed cases are systematically different from the fully observed cases; crucially, however, the difference is fully captured by x_i.

Figure 1 (c) provides an example of 'Missing Not At Random' (MNAR). This is characterised by dependence of the probability of missingness on both the partially and fully observed variables. Thus, in Figure 1 (c) π_i depends on both the fully observed predictor x_i and the partially observed outcome y_i. This means that the difference between fully and partially observed cases still depends on the missing values, even after taking x_i into account. Therefore it is necessary to make more structured assumptions about this relationship that go beyond the information contained in the data.

2.4 Selection Models

Independently of the setting analysed, if MCAR is not credible, there is one important issue that should always be considered in conducting analysis with missing data: one cannot definitively distinguish between MAR and MNAR models. The data alone do not provide all the information necessary to make this choice and, at the same time, different MNAR models can provide identical fits to the observed data. However, they may have quite different implications for the unobserved data, leading to different conclusions (Molenberghs et al., 2015). Therefore, it becomes crucial to explore the sensitivity of the results with respect to different missing data assumptions and quantify results' uncertainty. What is generally recommended is to set MAR as the reference assumption and then explore different MNAR departures. However, the base-case analysis should be primarily defined based on the available state of knowledge in the given setting. When informative missingness is thought to be the most realistic scenario, setting-specific MNAR assumptions should be set as the reference case, with suitably-defined departures being explored in *Sensitivity Analysis* (SA). Usually, SA for nonignorable/informative models is implemented through advanced statistical methods, which can explicitly model a MNAR mechanism. One popular technique is the *Selection Model* approach (Molenberghs et al., 2015; Daniels and Hogan, 2008; Mason et al., 2012).

To represent the application of selection models we consider a simple example. We assume a data set comprising a partially observed response variable y, the corresponding missing data indicator vector m, and a fully-observed covariate x. Under the SM approach, the joint distribution $p(y, m)$ is factored as the product of the marginal distribution $p(y)$ and the conditional distribution $p(m \mid y)$.

$$p(y, m \mid x, \boldsymbol{\theta}^{MoA}, \boldsymbol{\theta}^{MoM}) = p(y \mid x, \boldsymbol{\theta}^{MoA})p(m \mid y, x, \boldsymbol{\theta}^{MoM}) \tag{1}$$

where, $\boldsymbol{\theta}^{MoA}, \boldsymbol{\theta}^{MoM}$ are respectively the set of parameters associated with the MoA and that associated with the MoM, respectively. In Equation (1) we need to specify the complete data model for the response, so that the probability of nonresponse is modelled conditionally on the possibly unobserved outcomes. Model identifiability comes from some parametric assumption about $p(y)$. This will implicitly set up the relationship between parameters indexing the distribution of the observed and unobserved cases, together with some unverifiable assumptions on the specific form of $p(m \mid y)$.

To show how to build a selection model we use the simplified framework with only one partially observed outcome variable y. We then need to jointly model $p(y, m \mid x, \boldsymbol{\theta}^{MoA}, \boldsymbol{\theta}^{MoM})$ by specifying the two different modules associated with the observed and missingness processes (MoA and MoM), based on the selection model factorisation. While the MoA specification depends on the main parameters of interest and the setting-specific research question involved, the MoM can have a more generalised structure.

2.4.1 Model of Analysis

Under a Selection Model, we need to specify the form of the probability distribution assumed in the MoA to describe the uncertainty in the observed outcome variables, i.e. effects and costs. It is good practice to test a set of (more or less) plausible parametric models for the outcome data according to their characteristics and support.

In general terms, we can specify the vector of relevant parameters as $\boldsymbol{\theta}^{MoA} = (\mu, \alpha)$. Specifically, μ and α respectively represent a *location* parameter, which typically indicates the mean of the probability distribution, and a (set of) *ancillary* parameter(s), which describes the shape or variance of the distribution.

While it is possible for both μ and α to explicitly depend on some covariates $\boldsymbol{x}$, usually the formulation is simplified to assume that these only affect directly the location parameter. In addition, we typically use a linear formulation

$$\mu = \beta_0 + \sum_{j=1}^{J} \beta_j x_{ij} \tag{2}$$

to model the location parameter. In a full Bayesian setting, the parameters are directly modelled using a prior probability distribution, which is updated by the observed data into a posterior. It is this posterior distribution that is the object of the inferential process. For instance, we can assign the following priors to the regression coefficients: $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_J) \overset{iid}{\sim} \text{Normal}(\mu_\beta, \alpha_\beta)$. If no covariate data are provided then this would correspond to define a prior directly on the marginal mean parameter μ, whose specification will depend on the support for the given MoA distributions assumed.

Note that **missingHE** expands any categorical covariates to a set of dummy variables: so if a covariate has four categories, in line with R notation, **missingHE** considers three binary indicators. Thus the profile (0,0,0) indicates the first (reference) category, while the profiles (1,0,0), (0,1,0) and (0,0,1) indicate the second, third and fourth category, respectively. In **missingHE**, the number of covariates J depends on this full expansion of the design matrix.

By default, **missingHE** assumes $\mu_\beta = 0$ and $\alpha_\beta = 1000$. This amounts to specifying a “minimally informative" prior on the regression coefficients that determine the location parameter; in other words, we are not including strong prior information in this aspect of our model. When genuine prior knowledge is present, it is possible to modify these priors to encode the information in the model formulation. As for the ancillary parameter, the choice of prior depends on the specification of the probability distribution selected to model the data. Table 1 shows a summary of the models directly implemented in **missingHE**. In each, by default, we specify minimally informative priors on all the relevant parameters.

Data Model	Location Parameter	Ancillary Parameter	**missingHE** name
Shared Distributions			
$e_i; c_i \sim \text{Normal}(\mu, \alpha)$	Mean: $\mu \sim \text{Normal}(0, 10000)$	log-SD: $\alpha \sim \text{Uniform}(-5, 10)$	norm
Effect Distributions			
$e_i \sim \text{Beta}(\mu, \alpha)$	Mean: $\mu \sim \text{Uniform}(0, 1)$	SD: $\alpha \sim \text{Uniform}(0, \sqrt{\mu(1-\mu)})$	beta
Cost Distributions			
$c_i \sim \text{Gamma}(\mu, \alpha)$	Mean: $\mu \sim \text{Uniform}(0, 10000)$	SD: $\alpha \sim \text{Cauchy}(0, 2.5)\text{I}(0,)$	gamma

Table 1: A list of the distributions supported by **missingHE** for the effect (e_i) and cost (c_i) variables in the MoA with the default weakly informative prior forms assumed. The names on the right-hand side of the table represent the character names in **missingHE** notation to indicate the corresponding distribution.

To provide user-defined prior distributions for the location and ancillary parameters we need to create objects having names `mu.prior.e` and `alpha.prior.e` for the effects and `mu.prior.c` and `alpha.prior.c` for the costs, respectively. These objects must contain the hyperprior values defined by the user. For example, assuming to model both outcomes by a Normal distribution, the default priors for μ and α for the effects, for instance, can be overwritten by creating the two objects `mu.prior.e` and `alpha.prior.e` containing the desired values. Then, these values are supplied to the main function `run_model` as additional arguments (more on this later). In R this can be perfomed as follows:

```r
#hyperprior mean and standard deviation for the location parameter
a<-0
b<-1
```

```
mu.prior.e<-c(a,b)
#hyperprior lower and upper bound for the ancillary parameter
l<--5
u<-5
alpha.prior.e<-c(l,u)
```

The hyperprior values for both μ and α must be chosen according to the specific model structure assumed by **missingHE** as different outcome distributional assumptions may lead to different types of prior forms. The types of prior forms assumed for the location and ancillary parameters for alternative MoA specifications are shown in Table 1.

If covariate data are incorporated in the model, then the location paraemter μ is replaced by the expression of Equation (2) and hyperprior values should be supplied for each regression coefficient parameter $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_J)$. To notice that in this case all coefficient prior distributions are by default weakly-informative normal distributions except the baseline regression parameter β_0 that is considered and modelled equivalently to the marginal mean μ.

For example, if we assume to have two covariates that we want to include in the model, the user-provided hyperprior values for all the regression coefficient parameters can be specified by creating two objects having names `beta0.prior.e` and `beta.prior.e` for the effects and `beta0.prior.c` and `beta.prior.c` for the costs, respectively. These objects must contain the user-provided hyperprior values for the marginal mean and covariate specific coefficient parameters, respectively. For the latter, the hyperprior values supplied will be applied equivalently to all covariate-specific parameters.

Assuming again a bivariate normal distribution for (e_i, c_i), in R the default priors for the mean regression coefficient paraemters for the effects, for example, can be performed as follows:

```
#hyperprior mean and standard deviation for the marginal mean parameter
a<-0
b<-1
beta0.prior.e<-c(a,b)
#hyperprior mean and standard deviation for the regression coefficient parameters
c<-0
d<-1
beta.prior.e<-c(c,d)
```

2.4.2 Model of Missingness

A common procedure to build the MoM module under a Selection Model approach assigns a Bernoulli probability distribution to the missing data indicator $m_i \sim \text{Bernoulli}(\pi_i)$, where π_i is the parameter specifying the probability of y_i being missing, with either $y_i = e_i$ or $y_i = c_i$. At this point, we can model the missingness probability π_i in a variety of ways based on the assumptions we would like to make about the missing data mechanism. Here, we present a general model specification for a MNAR missingness mechanism that includes the MAR and MCAR assumptions as special cases.

$$\text{logit}(\pi_i) = \gamma_0 + \gamma_j x_{ij} + \delta y_i \tag{3}$$

where $\boldsymbol{\theta}^{MoM} = (\gamma_0, \gamma_j, \delta)$, with $j = 1, \ldots, J$ being the number of covariates. The "log odds" $\text{logit}(\pi)$ is computed as $\text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right)$.

The logistic regression is just a common transformation that is used to rescale the parameter space of π_i from $(0, 1)$ to $(-\infty, +\infty)$ so to include in the regression any variable upon which the missingness probability may depend. More specifically:

- γ_0 is a logistic regression baseline parameter that does not depend on any variable. When $\gamma_j = \delta = 0$, the model assumes MCAR, as missing values in the outcome y_i will be imputed independently on any variable in the MoA.
- γ_j represents the impact on the probability of missingness in y_i of the fully observed covariate x_{ij}. When $\delta = 0$, the model assumes MAR, as missing values in the outcome y_i will be imputed conditionally on x_{ij}.
- δ represents the impact on the probability of missingness in y_i of the possibly unobserved values in y_i. The model now assumes MNAR, as missing values in the outcome y_i will be imputed conditionally on themselves (given the extra information brought into the model to specify this conditional dependence).

As with respect to the prior distributions of the MoM parameters, as for the MoA parameters, they can all be provided by the users while the default values are chosen with the aim to limit the impact of possible prior informative content on posterior inferences. Table 2 shows a summary of the default settings for the logistic regression parameters based on the class of MoMs that can be indicated in **missingHE** (same structure is assumed for both effects and costs).

Missingness Model	Logistic Regression Parameter	**missingHE** name	
		effects	**costs**
MCAR	Baseline parameter: $\gamma_0 \sim \text{Logisitc}(0,1)$	gamma0.prior.e	gamma0.prior.c
MAR	Covariate(s) parameter: $\gamma_j \sim \text{Normal}(0,1)$	gamma.prior.e	gamma.prior.c
MNAR	MNAR parameter: $\delta \sim \text{Normal}(0,1)$	delta.prior.e	delta.prior.c

Table 2: A list of the different MoM parameter specifications assumed in **missingHE** and the default prior distributions assigned. The names on the right-hand side of the table represent the name to be used in **missingHE** to provide user-defined priors in a similar way to what previously shown for the location and ancillary MoA parameters.

For the MNAR parameter δ a similar prior to those used for γ_j is assumed for convenience. However, because such parameter expresses the dependence of the MoM on the missing data, then no actual “minimally informative" prior exist as there are no data able to inform its estimation. As a result, all choices may end up to have a quite significant impact on posterior results, and should be selected according to the context specific missingness information available. The exploration of plausible alternative prior values is necessary in a MNAR setting, as the only way to assess the robustness of posterior inferences to plausible alternative missingness scenarios. This is possible in **missingHE** by directly providing the values for the hyperparameter δ and compare how results change across different choices.

Selection Models allow to directly model the target distribution of the complete data. This has the advantage to straightforwardly formulate assumptions about the nonresponse mechanism. The drawback is how we can translate these assumptions into assumptions on the distribution of the missing data. Indeed, model identification depends on assumptions on the distribution of y_i (often difficult to check) and on the form of the missingness model (on which unverifiable assumptions have to be made). Moreover, once the model is specified, all parameters are identified by the observed data and model assumptions. The best way to assess the impact of missingness on results is to vary distributional assumptions in the MoA and the form of the MoM. Independently on the method chosen, it must be noticed that all models have to rely on some arbitrary assumptions in order to be identified. This means that inference is possible only after some unverifiable assumptions about the missingness process have been formulated. Within this context, sensitivity analysis becomes extremely important in order to assess the robustness of the conclusions by reporting the results obtained under a range of plausible assumptions.

2.5 Example

In the following, we use a running example to present the features of **missingHE**. Suppose that the user has a suitable dataset, perhaps obtained from a trial, in which data for each individual are recorded for the effectiveness and cost variables as well as for an arm indicator specifying whether the individual to whom the data refer belongs in the control or the active treatment arm of the trial. Of course, other variables may be observed, e.g. relevant covariates, such as sex, age or co-morbidity. Both outcome variables can have missing values while no unobserved values should be observed for the covariates as **missingHE** can only deal with missingness in the outcomes.

Assume that the data are available in the R workspace as a data-frame (say, `data`) that can be partially visualised using the following command

```
rbind(head(data),tail(data))
```

```
##              e         c t
## 1   0.5262802  88.84761 1
## 2   0.4168671  71.16754 1
## 3   0.7516885 110.13573 1
## 4   0.9547645  77.74878 1
## 5          NA        NA 1
## 6          NA        NA 1
## 245 0.2422857 116.66148 2
## 246 0.6418311 107.61227 2
## 247        NA        NA 2
## 248        NA        NA 2
## 249 0.5444913 127.72288 2
## 250        NA        NA 2
```

The dataset consists of 250 individuals in total, grouped in two arms (here arm = 1 indicates the controls and arm = 2 indicates the active treatment).

3 Bayesian Analysis via JAGS/BUGS

Cost-Effectvieness Models are implemented in **missingHE** using Bayesian specific program software languages, namely `BUGS` or `JAGS`, which are called from two corresponding R packages, **R2OpenBUGS** and **R2jags**. `BUGS` (Bayesian inference Using Gibbs Sampling) (Lunn et al., 2012) is a software package for performing Bayesian inference Using Gibbs Sampling. The user specifies a statistical model, of (almost) arbitrary complexity, by simply stating the relationships between related variables. The software includes an expert system, which determines an appropriate MCMC (Brooks et al., 2011) (Markov chain Monte Carlo) scheme (based on the Gibbs sampler) for analysing the specified model. `JAGS` (Plummer, 2010) (Just Another Gibbs Sampler) by Martyn Plummer is an open source program which was developed independently of the BUGS project. JAGS uses essentially the same model description language of `BUGS` but it has been completely re-written.

To illustrate how **missingHE** interfaces with these programs we continue the example in Section 2.5 and we assume for simplicity to assign a bivariate independent normal distribution to the effect and cost variables (e_i, c_i) in the MoA. With respect to the MoM we assume a MNAR structure for the effects while we keep a MCAR assumption for the costs in both treatment arms $t = 1, 2$. In **missingHE** we can implement the model using the `run_model` function in the following way.

```
set.seed(12345)
model<-run_model(data=data,dist_e="norm",dist_c="norm",type="MNAR_eff",stand=FALSE,ind=TRUE,
                 program="JAGS",forward=FALSE,prob=c(0.05,0.95),n.chains=2,n.iter=20000,
                 n.burnin=floor(20000/2),inits=NULL,n.thin=1,save_model=FALSE)
```

The `run_model` function takes as input four compulsory arguments.

- The first is the data frame `data` and contains the economic evaluation data to analyse in the format described in Section 2.5.
- The second is `type`, which indicates the class of missingness mechanism assumed by the user. The types allowed are ("MCAR","MAR","MNAR","MNAR_eff","MNAR_cost") and represent the different possible mechanism specifications. Because we assume a MNAR structure only for the effects, the class we need to specify for this example is "MNAR_eff" that defines a MNAR mechanism for the effects while keeping MCAR/MAR for the costs. For all classes, except from "MCAR", whenever `data` contains some covariate data, these are automatically included in the model through a mean linear regression in the MoA and added to the logistic regression in the MoM, after the corresponding covariate specific coefficients have been generated.
- The third and fourth inputs are the distributions assumed for the effect and cost variables in the MoA, `dist_e` and `dist_c`, respectively. The distributions that are available in **missingHE** and their parameterisations are presented in Table 1. In this example we assume a normal distribution for both outcomes, indicated in **missingHE** by the string "norm".

The other arguments are optional and are related to different aspects of the model. Among the most important optional arguments there are:

- `stand` (valid only for bivariate normal) specifies whether outcome variables should be modelled on their standardised (`TRUE`) or natural scale (`FALSE`). If scaled, variables are standardised using the corresponding sample mean and standard error so to have scaled mean of 0 and standard error of 0.5, following Gelman's recommendations (Gelman, 2006). The default value is `FALSE`.
- `ind` (valid only for bivariate normal) specifies whether outcome variables are modelled independetly (`TRUE`) or jointly (`FALSE`). The default value is `TRUE`.
- `transf` (valid only for beta and gamma distribution) specifies on which scale effect or cost variables are modelled. If no values is specified, outcomes are modelled on their natural scale. Alternative choices are the log-scale for costs (`log`) and the logit scale for effects (`logit`).
- `program` specifies which software program to use to run the model, i.e. `JAGS` for JAGS or `BUGS` for OpenBUGS.
- `forward` specifies whether the model should be run in forward sampling mode (`TRUE`), i.e. sampling from the prior without considering the data, or in standard samping mode (`FALSE`). The default value is `FALSE`.
- `prob` must be a vector of length two containing the lower and upper bounds for the credible intervals associated with the posterior distribution of the imputed outcome data. Default values specify a 95% CI.
- `n.chains` defines the number of Markov chains to use in the MCMC sampler.
- `n.burnin` defines the warmup phase for the MCMC iterations that is discarded.
- `n.iter` defines the number of total MCMC iterations to use per each chain (included the burnin).
- `n.thin` defines the thinning rate.
- `inits` is a list with `n.chains` elements; each element of the list is itself a list of starting values for the model, or a function creating (possibly random) initial values. If `inits=NULL`, either `BUGS` or `JAGS` will automatically generate the initial values for the parameters.
- `save_model` specifies whether the `txt` file containing the `BUGS` model should (`TRUE`) or should not (`FALSE`) be saved in the current working directory. The default value is `FALSE`.

Behind the scenes **missingHE** translates the model information provided in the inputs of `run_model` and writes a txt file in a form that can be read by either `BUGS` or `JAGS`. For the example considered, the corresponding model specification can be more easily represented as follows.

$$\begin{aligned}
&\textbf{Model of Analysis (MoA)} \\
e_{it} &\sim \text{Normal}(\mu^e_t, \sigma^e_t) \\
c_{it} &\sim \text{Normal}(\mu^c_t, \sigma^c_t) \\
&\textbf{Model of Missingness (MoM)} \\
m^e_{it} &\sim \text{Bernoulli}(\pi^e_{it}) \\
\text{logit}(\pi^e_{it}) &= \gamma^e_0 + \delta^e e_{it} \\
m^c_{it} &\sim \text{Bernoulli}(\pi^c_{it}) \\
\text{logit}(\pi^c_{it}) &= \gamma^c_0
\end{aligned}$$

In the MoA, e_{it} and c_{it} respectively represent the effectiveness and cost variables, for each treatment arm $t = 1, 2$ and individual $i = 1, \ldots, N_t$. Both variables are assumed to be independently normally distributed with mean and standard deviation parameters indicated by (μ^e_t, σ^e_t) and (μ^c_t, σ^c_t), respectively. In the MoM the missingness indicators for the outcome variables m^e_{it} and m^c_{it} are given a Bernoulli distribution whose parameters π^e_{it} and π^c_{it} are functions of other MoM parameters on the log-odds scale. Specifically, since we assumed a MNAR mechanism for the effects, the corresponding missingness probabilities in both arms are a function of the baseline parameter γ^e_0 and the MNAR parameter δ^e. Conversely, given the MCAR assumption for costs, the corresponding MoM specification inlcudes only the baseline parameter γ^c_0.

Prior distributions are assigned to both MoA and MoM parameters in a way that is consitent with the description in Section 2.4.1 and Section 2.4.2, respectively. These can be represented as follows.

$$\begin{aligned}
&\textbf{Prior Distributions (MoA)} \\
\mu^e_t &\sim \text{Normal}(0, 1000) \\
\mu^c_t &\sim \text{Normal}(0, 1000) \\
\log(\sigma^e_t) &\sim \text{Uniform}(-5, 10) \\
\log(\sigma^c_t) &\sim \text{Uniform}(-5, 10) \\
&\textbf{Prior Distributions (MoM)} \\
\gamma^e_0 &\sim \text{Logistic}(0, 1) \\
\delta^e_0 &\sim \text{Normal}(0, 1) \\
\gamma^c_0 &\sim \text{Logistic}(0, 1)
\end{aligned}$$

In the MoA, weakly informative normal prior distributions centred at 0 and with larger variances are assigned to mean paraemters, while uniform distributions are assigned to standard deviation parameters on the logarithm scale. In the MoM, standard logistic prior distributions are assigned to the baseline parameters for both mechanisms to minimise the informative impact on the probability scale. For the effects, however, the assumed structure identified by the MNAR parameter δ^e highly depends on the corresponding hyper prior values chosen. By default values are similar to those used for the mean parameters in the MoA but, as explained in Section 2.4.2, users should explore plausible priors according to the available information so to assess the robustness of the results to alternative MNAR structures.

This can be performed in **missingHE** by providing the prior hyperparameter values desired by the users. For instance, for the bivariate normal model example considered, priors for mean effect and cost parameters can be modified by adding inside the function `run_model` the optional argument `prior` that should be defined as a list. This list must contain a number of vectors equivalent to the number of parameter classes whose prior we desire to change. These vectors, in turn, must contain the hyperparameter values we want to specificy and must comply with the prior forms assumed by **missingHE** based on the type of MoA and MoM structure assumed.

Returning to the example, if we want to change the priors of the mean location parameter μ^e and for the MNAR parameter δ^e we can do it by creating two objects named `mu.prior.e` and `delta.prior.e` that contain the hyperprior values desired. In R this can be performed as follows:

```
#hyperprior mean and standard deviation for the effect location parameter
a<-0
b<-1
mu.prior.e<-c(a,b)
#hyperprior mean and standard deviation for the MNAR effect parameter
c<-0
d<-1
delta.prior.e<-c(c,d)
```

Then, we must include these vectors inside a list object that is passed as an additional argument, called `prior`, to the function `run_model`. It is important that such list object must contain vector objects whose **string** names match those accepted by **missingHE**. Using the hyperprior vectors created before, we can proceed as follows:

```
prior<-list("mu.prior.e"=mu.prior.e,"delta.prior.e"=delta.prior.e)
model<-run_model(data=data,dist_e="norm",dist_c="norm",type="MNAR_eff",stand=FALSE,ind=TRUE,
                 program="JAGS",forward=FALSE,prob=c(0.05,0.95),n.chains=2,n.iter=20000,
                 n.burnin=floor(20000/2),inits=NULL,n.thin=1,save_model=FALSE,prior=prior)
```

Executing the command above creates an object `model` in the class `missingHE`, in which the results of the economic analysis are stored for the given MoA-MoM specification considered. The usual R command

```
names(model)
```

returns the names of the elements in the list

```
## [1] "data_set"     "model_output" "cea"          "type"
## [5] "model_class"
```

The objects `data_set`, `model_output` and `cea` are themselves lists that contain different elements related to the data provided, the model results and the economic analysis, respectively. For example, the elements in the first object can be accessed using the standard R notation `model$data_set[]` (i.e. using double square brackets) and can be inspected typing the command

```
## [1] "effects"                      "costs"
## [3] "N in reference arm"           "N in comparator arm"
## [5] "N observed in reference arm"  "N observed in comparator arm"
## [7] "N missing in reference arm"   "N missing in comparator arm"
## [9] "covariates"
```

These are merely the data related to the inputs given to the function `run_model`, such as effect and cost data, total number of individuals in each arm, number of observed and unobserved individuals in each arm and possible covariate data. The other elements of the object `model` are

- `model_output` is a list storing the output of the `BUGS` model. Depending on the type of model, the results shown in this list can vary as they contain the posterior samples of the parameters of interest either in the MoA and MoM or both (according to the MoA-MoM structure assumed). In the current example since a MNAR mechanism is assumed only for the effects, `model_output` contains the posterior samples for δ^e but does not contain any MoM parameters associated with the costs for which an ignorable mechanism was assumed (MCAR). `model_output` also contains a summary of the `BUGS` model that is taken directly from the output of the functions in the package **R2OpenBUGS** or **R2jags** and which contain all the information related to the model.

- `cea` is another list that stores the output of the economic evaluation based on the mean posterior samples of the mean effect and cost parameters and which is implemented using the functions in the package

BCEA. This object can be analysed using tailored functions of **BCEA** to visually represent standard CEA outputs such as the Cost-Effectiveness Plane (CEP) (Black, 1990) and the Cost-Acceptability Curve (CEAC) (Van Hout et al., 1994).

- `type` and `model_class` are string variables that specify the type of mechanism assumed and sampling implemented, respectively.

3.1 Model Convergence Assessment

As with any MCMC estimation, it is important to thoroughly assess convergence. The package **missingHE** contains the function `diagnostic_checks` to visualise the model output and assess convergence. Different diagnostic tools and plots for the model parameters are taken from the package **ggmcmc** and **mcmcplots** and are displayed using functions from **ggplot2** according to the inputs provided by the user. For excample, considering the model output generated in `model` for the current example, we can visually represent via histograms the posterior samples for the mean effect parameters in the two arms in the following way.

```
check<-diagnostic_checks(x=model,type="histogram",param="mu.e",theme=NULL)
```

which displays the graph in Figure 2.

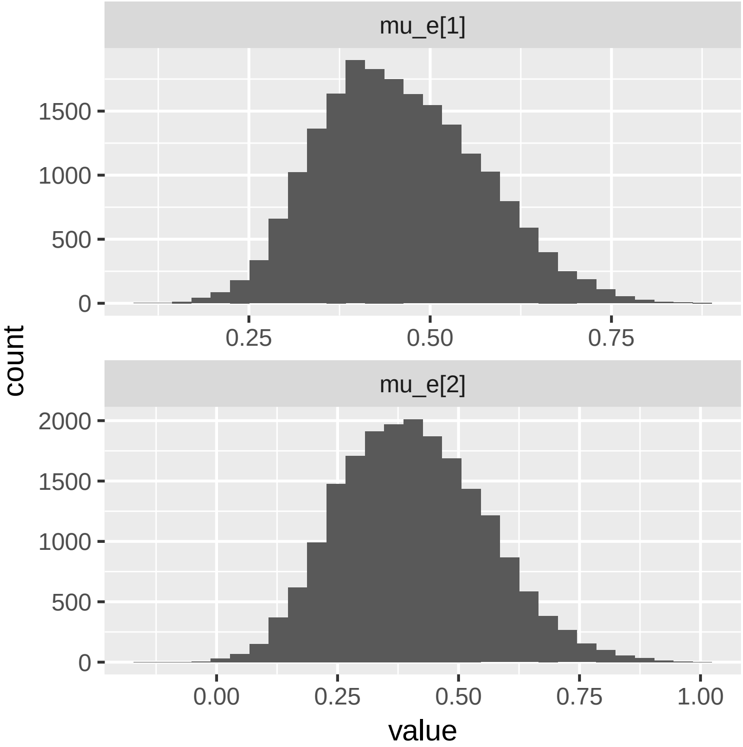


Figure 2: Checking model convergence using the **ggmcmc** package built-in facilities, for example through the inspection of the histogram plots by treatment arm for the mean effect parameters.

The function `diagnostic_checks` takes as compulsory input `x` that must be an object of class `misisngHE`, such as the object `model` generated by the function `run_model`. All other inputs are optional and are mainly used for selecting the parameters of interest and for graphics changes. Among these the most important are:

- `type` specifies the type of diagnostic tools to use for assessing convergence of the MCMC algorithm. A variety of plots are available such as histograms (`histogram`), density plots (`denplot`), traceplot

(traceplot), autocorrelation plots (acf), etc. The full list of all available types of plots can be found in the package manual. In addition, the class summary can be selected to display a summary of some of the most important diagnostic plots for each parameter selected.

- param specifies for which family of model parameters the diagnostic output should be displayed and must correspond to a string variable among a set of pre-defined choices. Specifically, the mean effect or cost parameters in the MoA can be assessed via the expressions "mu.e" and "mu.c", respectively. If a MNAR mechanism is assumed, then also the MoM parameters can be assessed; for instance, in our example, we can select the MNAR parameter for the MoM in the effect variables via the expression "delta.e". Only one family of parameters can be visualied at a time. A family of parameters is defined to be any group of parameters with the same name but different numerical values between square brackets (as beta[1], beta[2], etc). The list of all parameters that can be specified with the corresponding string names to be used in param can be found in the package manual. Alternatively, the set of all model parameters can be accessed by setting param="all", which also is the default value.
- theme is merely a graphical argument which modifies the pre-defnied background theme of the plots generated. Pre-defined themes are taken from the package **ggthemes** and must be indicated with corresponding expressions. Examples are "base", "calc", "economist", etc. For a full list of available themes see the package manual.

It is also possible to combine multiple graphs by running diagnistic_checks different times, setting different parameters to monitor, and save the plots in corresponding R objects. We can then combine different plots into a single one using the function grid.arrange from the **gridExtra** package (that should be loaded). For example, we can combine the density and trace plots for the mean effect parameters in the following way.

```r
require(gridExtra)
dens_eff<-diagnostic_checks(x=model,type = "denplot",param = "mu.e")
autoc_eff<-diagnostic_checks(x=model,type = "traceplot",param = "mu.e")
grid.arrange(dens_eff$plot, autoc_eff$plot, ncol=2)
```

which returns the graphs in Figure 3.

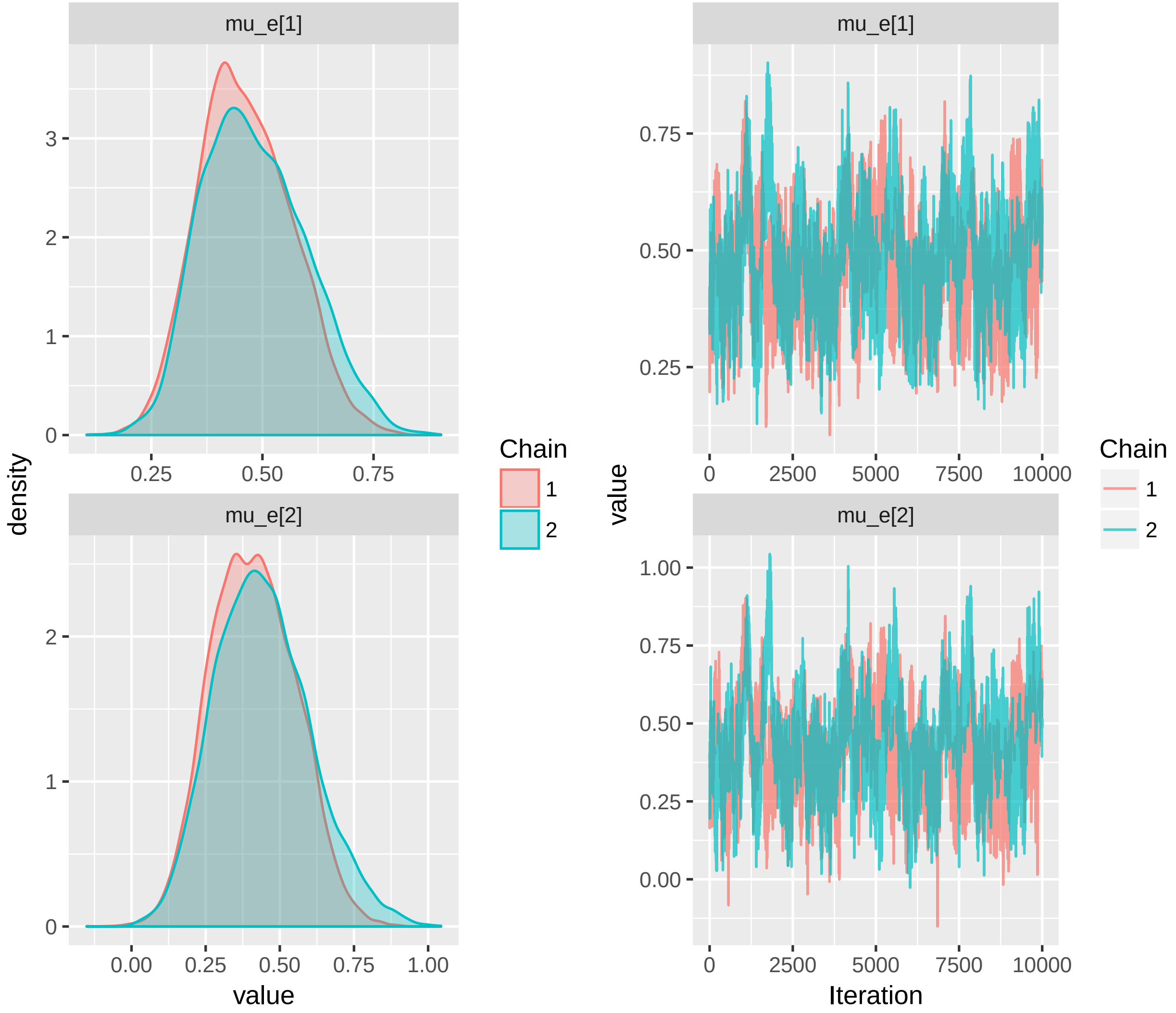


Figure 3: Combining different types of diagnostic check plots into a single one using the function `grid.arrange` in the package **gridExtra**. For example, here the density and trace plots for the mean effect parameters in the two arms are combined.

3.2 Summarising the Results from MissingHE

Objects in the class `missingHE` (such as `model` above) can access methods such as `summary` and `plot` that can be used to summarise the economic results and visually inspect how missing data have been imputed in the model analysed.

3.2.1 Missing Data Plots

Once the model have been estimated, we can visually inspect how missing data in the outcome variables have been imputed and compare them to the observed data. **MisisngHE** has a specialised function `plot` that can do this, by typing:

```
plot(x=model,class="scatter",outcome="all",theme="base")
```

which displays the graphs shown in Figure 4.

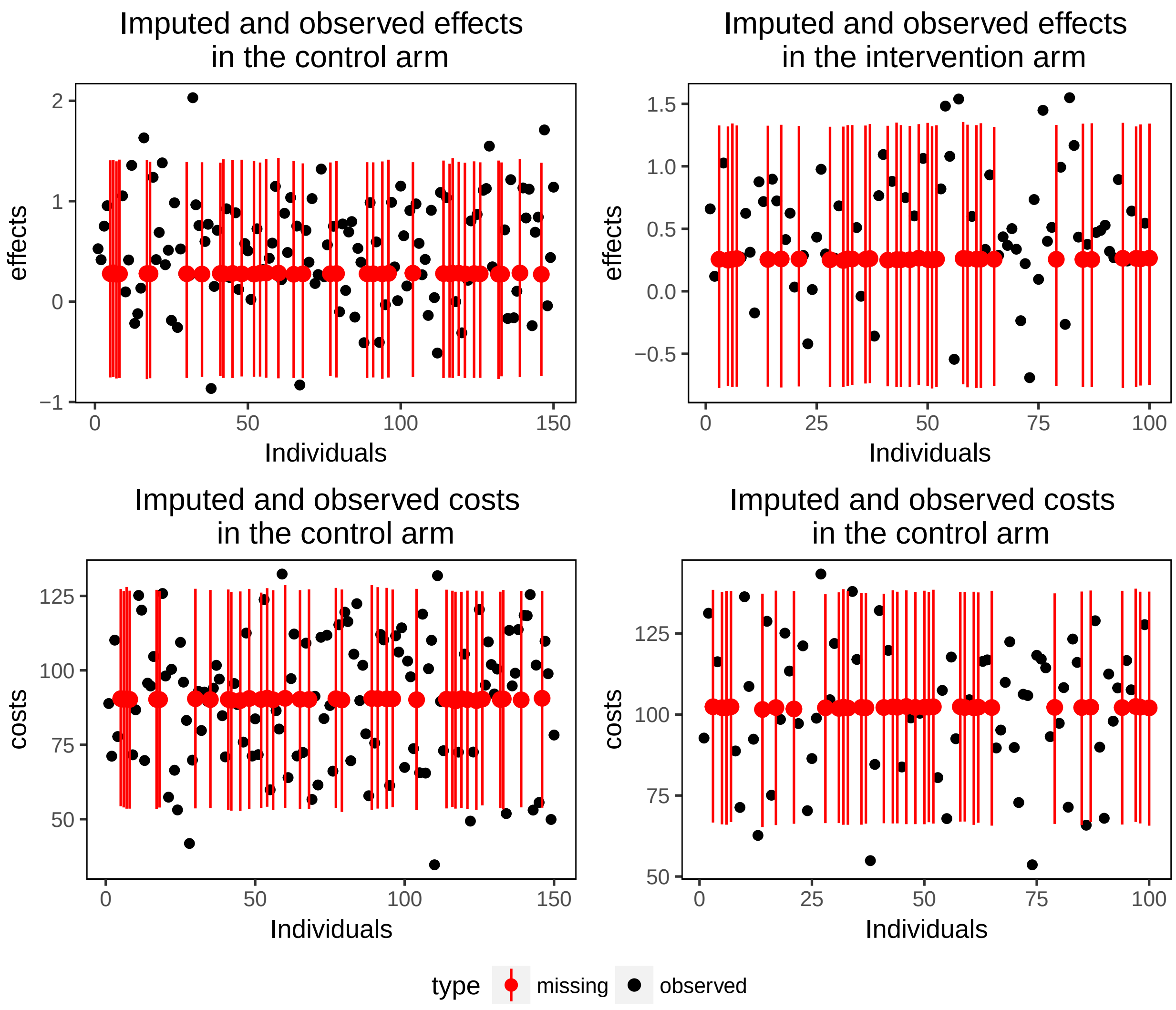


Figure 4: Comparing the observed and imputed effect and cost values in both treatment arms. Observed data are indicated with black dots while imputations are denoted by red dots and lines that define the credible intervals associated.

The only compulsory argument to be provided (`x`) is the `missingHE` object containing the results from the function `run_model`. All the other arguments are optional and are mainly related to the type of plot to be shown, which outcome and treatment arm to consider, and other graphics parameters. For a complete list of all arguments we refer to the package manual. Here we focus on two of the most important:

- `class` specifies the type of plot to be displayed. Two alternatives are available: “scatter” and “histogram”. Choosing “scatter” the observed and imputed values (evaluated at the posterior means) are shown in a scatter plot, with unobserved data also associated with lines representing their posterior credible intervals. By default these are the 95% CI but they can also be modified by changing the values for the upper and lower bounds using the `prob` argument in the function `run_model`. Setting “histogram” we compare the observed and missing values in a histogram plot and associate them with different colours.

- `outcome` specifies for which variable, either effects, costs or both, and for which treatment arm, either control, intervention or both, results should be visualised. For example, the plots only for the effects (costs) can be selected setting `outcome` equal to “effects” (“costs”), while the plots by treatment arm can be accessed setting `outcome` equal to “arm1” (control) or “arm2” (intervention). Plots for each combination of outcome and treatment group can also be specified using the string names "effects_arm1","costs_arm1","effects_arm2" or "costs_arm2". By default all plots are displayed using the string name "all".

- `theme` modifies the graphical output according to some pre-specified themes similarly to what shown for the `diagnostic_checks` function.

3.3 Summary CEA results

Results from the economic evaluation performed by **missingHE** in the background can be summarised in a tabular form using the specialised function `summary` by typing:

```
summary<-summary(x=model)
```

which returns the following table:

```
##
##  Cost-effectiveness analysis summary
##
##  Comparator intervention: intervention 1
##  Reference intervention: intervention 2
##
##  Model of Analysis (MoA) parameter estimates under MNAR_eff assumption
##
##  Comparator intervention
##                mean    sd     LB     UB
## mean effects  0.473 0.111  0.306  0.665
## mean costs   90.289 2.082 86.924 93.751
## sd effects    0.568 0.051  0.494  0.661
## sd costs     22.214 1.472 19.955 24.762
##
##  Reference intervention
##                 mean    sd     LB      UB
## mean effects   0.427 0.152  0.193   0.687
## mean costs   102.193 2.644 97.845 106.484
## sd effects      0.53 0.072  0.434   0.667
## sd costs      21.551 1.933 18.642   24.96
##
##  Incremental results
##                   mean    sd     LB     UB
## delta effects   -0.046 0.093 -0.202  0.106
## delta costs     11.904 3.364  6.367 17.385
## ICER          -260.259
```

The only argument that must be provided is the object `x` containing the results from the function `run_model`. Information is reported only for the main parameters of interest in the MoA for performing the economic evaluation, that is the mean and standard deviation parameters for both outcomes and treatment groups. In addition, the incremental mean results are provided at the bottom of the table, denoted with `delta effects` and `delta costs` respectively, with also the value of the ICER. Results are summarised in terms of posterior mean, standard deviation and 95% credible intervals for each parameter

A series of useful functions are included in the package **BCEA** that summarise the economic evaluation results obtained from the posterior samples of the mean effect and cost paraemters. Figure 5 shows as an example the CEP and CEAC plots obtained from applying the respective functions `ceac.plot` and `ceplane.plot` to the `BCEA` object contained in `model` and that can be accessed via `model$cea`. The R commands used to generate and combine these plots are the following.

```
require(ggplot2)
require(BCEA)
ceac.plot(model$cea,graph = "ggplot2")+ggtitle("CEAC")
ceplane.plot(model$cea,graph = "ggplot2")+ggtitle("CEP")
```

and the resulting output is given in Figure 5

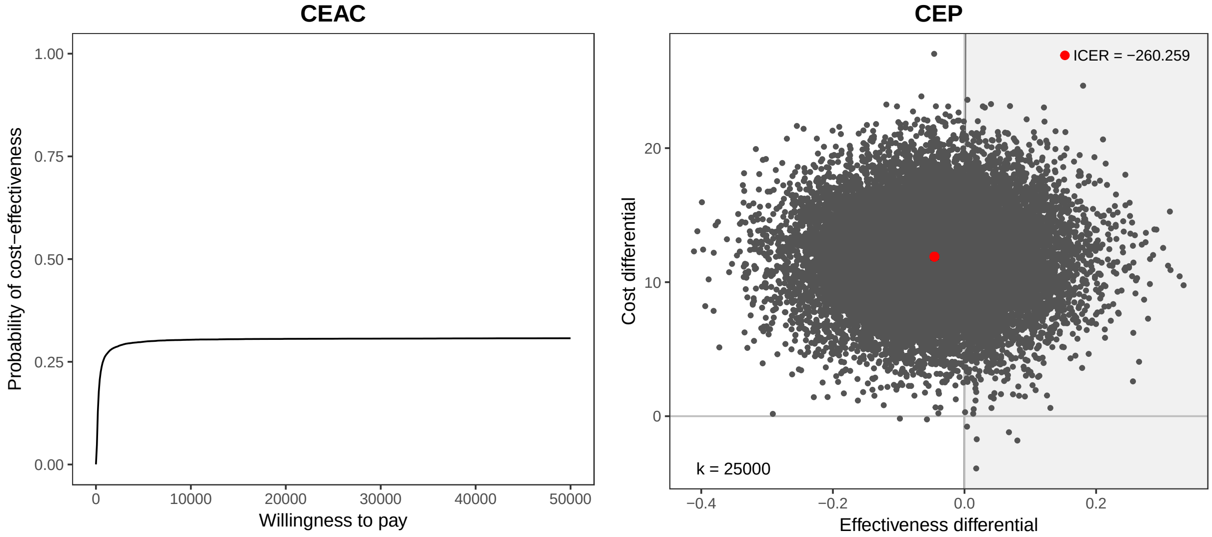


Figure 5: Cost Effectiveness Acceptability Curve (CEAC) and Cost-Effectiveness Plane (CEP) obtained using respectively the function `ceac.plot` and `ceplane.plot` in the package **BCEA** and applied to the model results contained in the object model

References

Arnold, J., Daroczi, G., Werth, B., B., W., Kunst, J., Auguie, B., Rudis, B., and Wickham, H. (2017). Package ‘ggthemes’. https://cran.r-project.org/web/packages/ggthemes/.

Auguie, B. and Antonov, A. (2016). Package ‘gridExtra’. https://cran.r-project.org/web/packages/gridExtra/.

Baio, G. (2013). *Bayesian Methods in Health Economics*. Chapman and Hall/CRC, University College London London, UK.

Baio, G., Berardi, A., and Heath, A. (2016). Package ‘BCEA’. https://cran.r-project.org/web/packages/BCEA/.

Black, W. (1990). A graphic representation of cost-effectiveness. *Medical Decision Making*, 10:212–214.

Briggs, A. and Gray, A. (1999). Handling uncertainty when performing economic evaluation of healthcare interventions. *Health Technology Assessment*, 3:1–134.

Briggs, A., Sculpher, M., and Claxton, K. (2006). *Decision modelling for health economic evaluation*. OUP, Oxford, UK.

Brooks, S., Gelman, A., Jones, G., and Meng, X. (2011). *Handbook of Markov Chain Monte Carlo*. CRC press.

Curtis, S., Goldin, I., and Evangelou, E. (2015). Package ‘mcmcplots’. https://cran.r-project.org/web/packages/mcmcplots/.

Daniels, M. and Hogan, J. (2008). *Missing Data in Longitudinal Studies: Strategies for Bayesian Modeling and Sensitivity Analysis*. Chapman and Hall, New York.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 1:515–533.

Gelman, A., Sturtz, S., and Ligges, U. (2017). Package 'R2OpenBUGS'. https://cran.r-project.org/web/packages/R2OpenBUGS/.

Lunn, D., Jackson, C., Best, N., Thomas, A., and Spiegelhalter, D. (2012). *The BUGS book: A practical introduction to Bayesian analysis*. CRC press.

Marin, X. (2016). Package 'ggmcmc'. https://cran.r-project.org/web/packages/ggmcmc/.

Mason, A., Richardson, S., Plewis, I., and Best, N. (2012). Strategy for modelling nonrandom missing data mechanisms in observational studies using bayesian methods. *Journal of Official Statistics*, 28:279–302.

Molenberghs, G., Fitzmaurice, G., Kenward, M., Tsiatis, A., and Verbeke, G. (2015). *Handbook of Missing Data Methodology*. Chapman and Hall, Boca Raton, FL.

Plummer, M. (2010). JAGS: Just Another Gibbs Sampler. http://www-fis.iarc.fr/~martyn/software/jags/.

Rubin, D. (1987). *Multiple Imputation for Nonresponse in Surveys*. John Wiley and Sons, New York,USA.

Van Hout, B., Al, M., Gordon, G., Rutten, F., and Kuntz, K. (1994). Costs, effects and c/e-ratios alongside a clinical trial. *Health Economics*, 3:309–319.

Wickham, H. and Chang, W. (2016). Package 'ggplot2'. https://cran.r-project.org/web/packages/ggplot2/.

YS., S. and Yajima, M. (2015). Package 'R2jags'. https://cran.r-project.org/web/packages/R2jags/.