Methods for addressing missing data in health economic evaluations

XXXIX Spanish Health Economics Meeting

Pre-conference worshop: Solutions

11 June 2019

Exercise 1: Install missingHE and inspect the data

a-b Open `Rstudio` and proceed to install the **missingHE** package using the command

```
> install.packages("missingHE")
```

Next, load the package using

```
> library(missingHE)
```

The object `MenSS` can be now accessed. We can use the command `str` to obtain summary information about the number and type of variables included in this dataset.

```
> str(MenSS)
'data.frame': 159 obs. of  8 variables:
 $ id        : int  1 2 3 4 5 6 7 8 9 10 ...
 $ u.0       : num  0.725 0.848 0.848 1 0.796 ...
 $ e         : num  NA 0.924 NA NA NA 0.943 NA NA NA 0.631 ...
 $ c         : num  NA 0 NA NA NA 0 NA NA NA 516 ...
 $ age       : int  23 23 27 27 18 25 48 30 24 27 ...
 $ ethnicity : Factor w/ 2 levels "0","1": 1 1 2 1 1 2 2 2 2 1 ...
 $ employment: Factor w/ 2 levels "0","1": 1 2 1 1 1 2 2 2 2 2 ...
 $ t         : int  1 1 1 1 1 1 1 1 1 1 ...
```

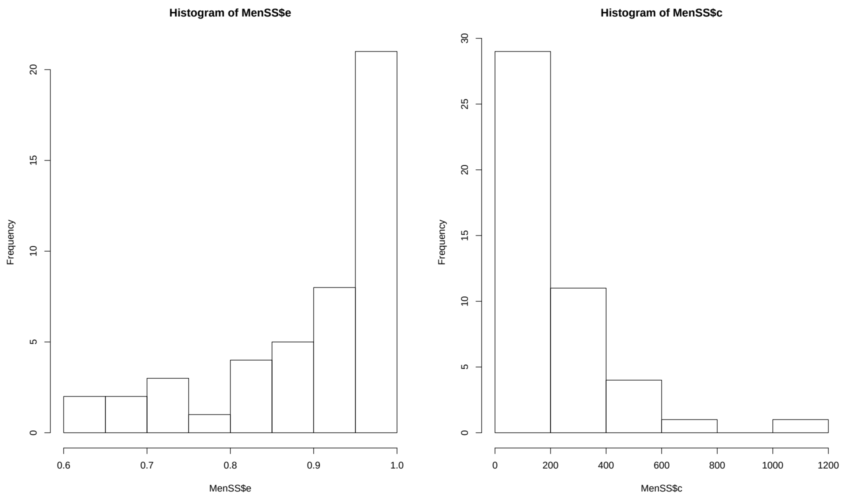
There are 159 individuals in the trial and for each of them, economic data are available for the QALYs (e), total costs (c) and baseline utilities (u_0). Three other baseline covariates are included in the dataset: age (continuous), ethnicity and employment (binary), while t denotes the treatment indicator which assigns individuals to either the control ($t = 1$) or intervention ($t = 2$) group..

c A quick summary of the dataset can be obtained by typing

```
> summary(MenSS)
       id             u.0               e                 c         
 Min.   :  1.0   Min.   :0.0020   Min.   :0.6148   Min.   :   0.0  
 1st Qu.: 40.5   1st Qu.:0.8480   1st Qu.:0.8486   1st Qu.:   0.5  
 Median : 80.0   Median :0.8819   Median :0.9371   Median : 143.0  
 Mean   : 80.0   Mean   :0.8819   Mean   :0.9031   Mean   : 200.3  
 3rd Qu.:119.5   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 329.8  
 Max.   :159.0   Max.   :1.0000   Max.   :1.0000   Max.   :1039.0  
                                  NA's   :113      NA's   :113     
      age         ethnicity employment       t        
 Min.   :16.00   0:76      0: 49      Min.   :1.000  
 1st Qu.:23.00   1:83      1:110      1st Qu.:1.000  
 Median :27.00                        Median :2.000  
 Mean   :29.38                        Mean   :1.528  
 3rd Qu.:33.50                        3rd Qu.:2.000  
 Max.   :67.00                        Max.   :2.000  
```

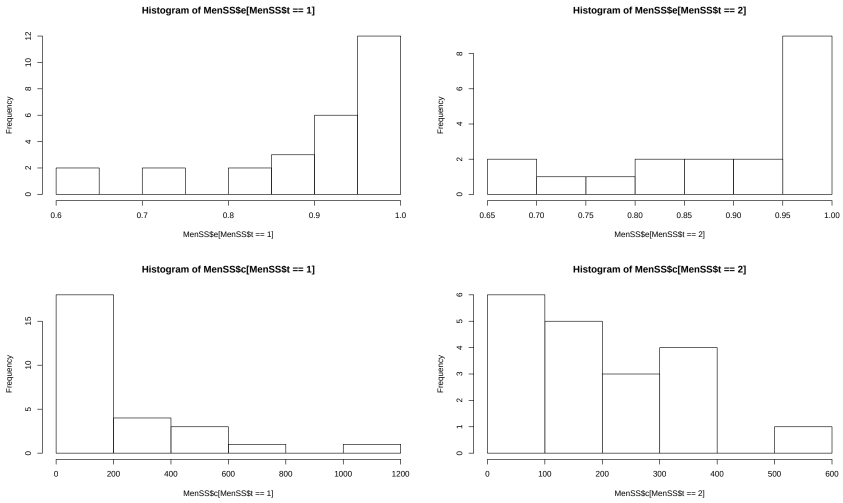
All variables, with the exception of e and c, are fully observed. There is a total of 113 individuals with missing QALYs and cost data (i.e. about $\approx 70\%$). From the value of the summary statistics for e and c we can see that the observed QALYs lie between [0.6,1] and are negatively skewed (mean $<$ median), while costs are defined between [0,1000] and are positively skewed (mean $>$ median). In addition, there are individuals who are associated with some structural values in both the QALYs (1) and costs (0). We can inspect the distribution of the two outcomes using histograms, e.g. using the following commands

```
> par(mfrow=c(1,2))
> hist(MenSS$e)
> hist(MenSS$c)
```



The degree of skewness in the empirical distributions of the two outcomes appears to be relatively high with a substantial spike at the boundaries of the value range. You can also display the distributions of the variables by treatment group using the commands

```
> par(mfrow=c(2,2))
> hist(MenSS$e[MenSS$t==1])
> hist(MenSS$e[MenSS$t==2])
> hist(MenSS$c[MenSS$t==1])
> hist(MenSS$c[MenSS$t==2])
```



Exercise 2: Selection models

a Follow the script and fit the bivariate Normal model using the `selection` function

```
> set.seed(123)
> NN.sel=selection(data = MenSS, model.eff = e ~ u.0, model.cost = c ~ e,
+  model.me = me ~ 1, model.mc = mc ~ 1, type = "MAR", n.chains = 2,
+  n.iter = 10000, n.burnin = 1000, dist_e = "norm", dist_c = "norm",
+  save_model = TRUE)
```

b The `JAGS` model generated in the file `selection.txt` by the commands above can be represented as follows. First, a marginal Normal distribution for the effectiveness is assumed

$$e_i \sim \text{Normal}(\phi_{iet}, \sigma_{et}),$$

where the individual mean response (QALYs) is modelled using a linear regression as a function of the baseline utility

$$\phi_{iet} = \alpha_{0t} + \alpha_{1t} u_{i0t}.$$

In the `JAGS` code, Normal distributions are defined in terms of precision τ_{et} rather than the variance, where $\tau_{et} = \frac{1}{\sigma^2_{et}}$. The linear predictor simply translates the conditional regression for ϕ_{iet}, while the marginal mean of the QALYs μ_{et} is calculated by replacing the baseline utilities with their mean values

$$\mu_{et} = \alpha_{0t} + \alpha_{1t}\bar{u}_{0t}.$$

Next, a similar approach is used to model the costs

$$c_i \sim \text{Normal}(\phi_{ict}, \sigma_{ct}),$$

where the mean is specified as the linear predictor

$$\phi_{ict} = \beta_{0t} + \beta_{1t}(e_i - \mu_{et}).$$

Because the covariate included in this model is centered, the marginal mean costs corresponds to the intercept term, i.e. $\mu_{ct} = \beta_{0t}$.

The models for the missing data indicators is specified using Bernoulli distributions,

$$m_{ie} \sim \text{Bernoulli}(\pi_{et}) \quad \text{and} \quad m_{ic} \sim \text{Bernoulli}(\pi_{ct})$$

where the probability of missingness is modelled on the logit scale

$$\text{logit}(\pi_{et}) = \gamma_{et} \quad \text{and} \quad \text{logit}(\pi_{ct}) = \gamma_{ct}.$$

Since no covariates are included in the models, the probabilities of missingness depend only on the random terms γ_{et} and γ_{ct}.

The `JAGS` code is replicated for the two treatment groups and by default it specifies vague prior distributions on all model parameters:

- $\boldsymbol{\alpha} = (\alpha_{0t}, \alpha_{1t}) \overset{iid}{\sim} \text{Normal}(0, 0.0000001)$
- $\boldsymbol{\beta} = (\beta_{0t}, \beta_{1t}) \overset{iid}{\sim} \text{Normal}(0, 0.0000001)$
- $\text{log}\boldsymbol{\sigma} = (\text{log}\sigma_{et}, \text{log}\sigma_{ct}) \overset{iid}{\sim} \text{Uniform}(-5, 10)$, which induce the priors on $\boldsymbol{\sigma}$ and then on $\boldsymbol{\tau}$
- $\boldsymbol{\gamma} = (\gamma_{et}, \gamma_{ct}) \overset{iid}{\sim} \text{Logistic}(0, 1)$

The `JAGS` code maps these assumptions directly and also adds some lines to derive the marginal missingness probabilities (p_{et}, p_{ct}) and the log-likelihood for each node which is explicitly modelled. These are then used by **missigHE** to compute different measures of model fit.

c We can visualise key posterior summaries of all the parameters in the model by typing

```
> print(NN.sel)
##                 mean      sd      2.5%    97.5% Rhat n.eff
## alpha[1,1]     0.185   0.140    -0.090    0.461    1 18000
## alpha[2,1]     0.782   0.151     0.482    1.079    1 18000
## alpha[1,2]     0.665   0.075     0.515    0.814    1 18000
## alpha[2,2]     0.286   0.087     0.112    0.458    1 18000
## beta[1]      237.773  51.689   137.070  342.027    1 18000
## beta[2]      185.971  41.381   103.167  266.798    1 18000
## beta_f[1]   -977.668 420.401 -1796.513 -155.620    1 18000
## beta_f[2]  -183.154 363.413  -889.444  529.215    1 18000
## deviance    903.701   5.912   894.266  917.406    1 18000
## gamma_c[1]    0.566   0.238     0.106    1.038    1 13000
## gamma_c[2]    1.212   0.257     0.725    1.731    1 10000
## gamma_e[1]    0.568   0.240     0.103    1.045    1  9500
## gamma_e[2]    1.210   0.253     0.728    1.721    1 18000
## mu_c[1]     237.773  51.689   137.070  342.027    1 18000
## mu_c[2]     185.971  41.381   103.167  266.798    1 18000
## mu_e[1]       0.874   0.017     0.841    0.906    1 18000
## mu_e[2]       0.917   0.022     0.874    0.961    1 18000
## p_c[1]        0.636   0.054     0.526    0.738    1 15000
## p_c[2]        0.767   0.045     0.674    0.849    1 11000
## p_e[1]        0.637   0.055     0.526    0.740    1 17000
## p_e[2]        0.767   0.045     0.674    0.848    1 18000
## s_c[1]      243.422  35.758   185.450  324.352    1 18000
## s_c[2]      170.871  31.568   122.509  246.001    1 18000
## s_e[1]        0.081   0.012     0.062    0.107    1 18000
## s_e[2]        0.092   0.017     0.066    0.131    1  2700
```

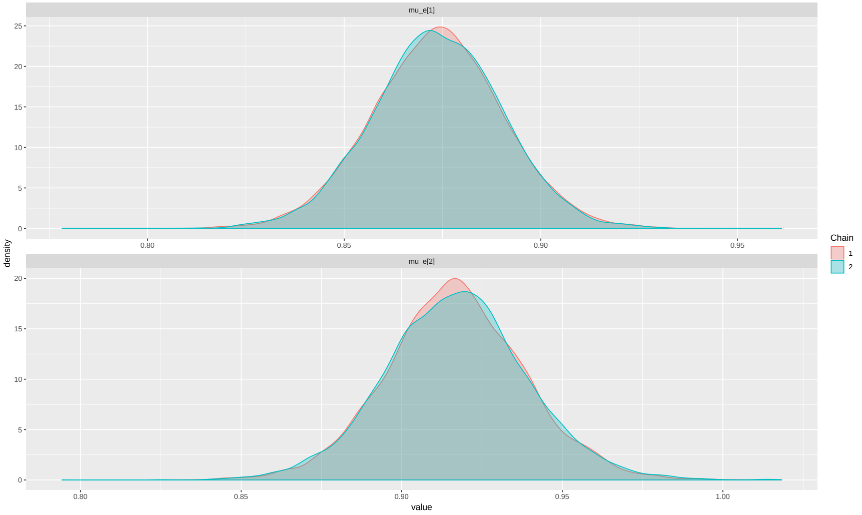
We can also access selected summary statistics for the mean QALY and cost parameters (μ_{et}, μ_{ct}), such as mean, sd, 2.5- and 97.5% quantiles as well as the convergence statistics, by using the following command

```
> NN.sel$model_output$summary[grep("mu",rownames(NN.sel$model_output$summary)),
+  c("mean","sd","2.5%","97.5%","Rhat","n.eff")]
##            mean     sd    2.5%   97.5%  Rhat n.eff
## mu_c[1] 237.773 51.689 137.070 342.027 1.001 18000
## mu_c[2] 185.971 41.381 103.167 266.798 1.001 18000
## mu_e[1]   0.874  0.017   0.841   0.906 1.001 18000
## mu_e[2]   0.917  0.022   0.874   0.961 1.001 18000
```

which shows the selected summary statistics for all the nodes whose name contains the keyword `mu` (this is done using the `grep` function - see `help(grep)` for more details).

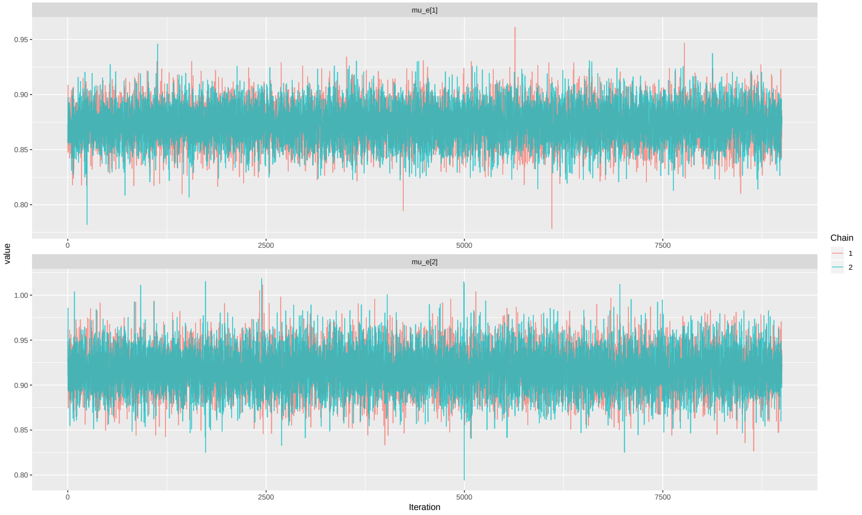
d Different types of diagnostic plots can be selected using the `type` argument in the `diagnostic` function. Two of the most popular diagnostic graphs are the density and trace plots. The former correspond to smoothed histograms of the posterior samples of the parameters in each chain, while the latter plot the value of the parameters at each iteration for each chain. For example, we can display the density plots for the mean QALYs in both treatment groups by setting the arguments `type="denplot"` and `param="mu.e"`.

```
> diagnostic(NN.sel, type = "denplot", param = "mu.e")
```



Similarly, we can obtain the trace plots for the same parameters by setting the argument `type="traceplot"`

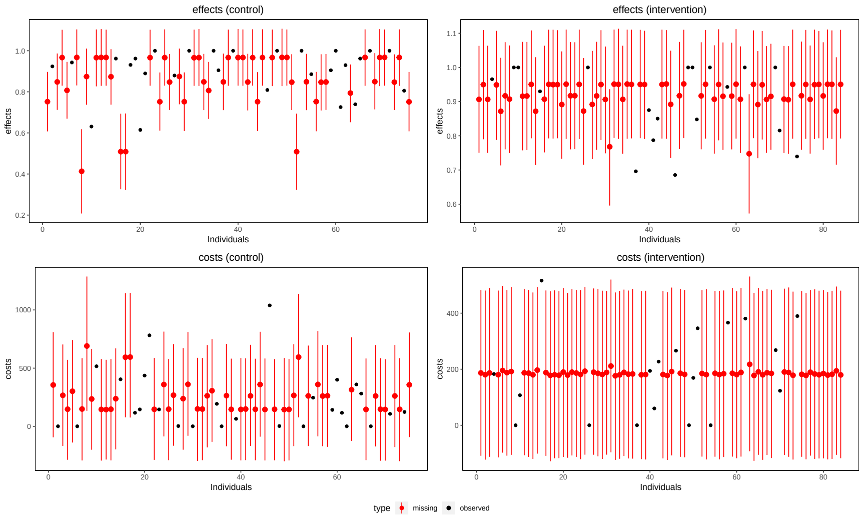
```
> diagnostic(NN.sel, type = "traceplot", param = "mu.e")
```



Both types of graphs do not show evidence of any issue in the convergence of the MCMC algorithm for these parameters and suggest a good mixing of the chains.

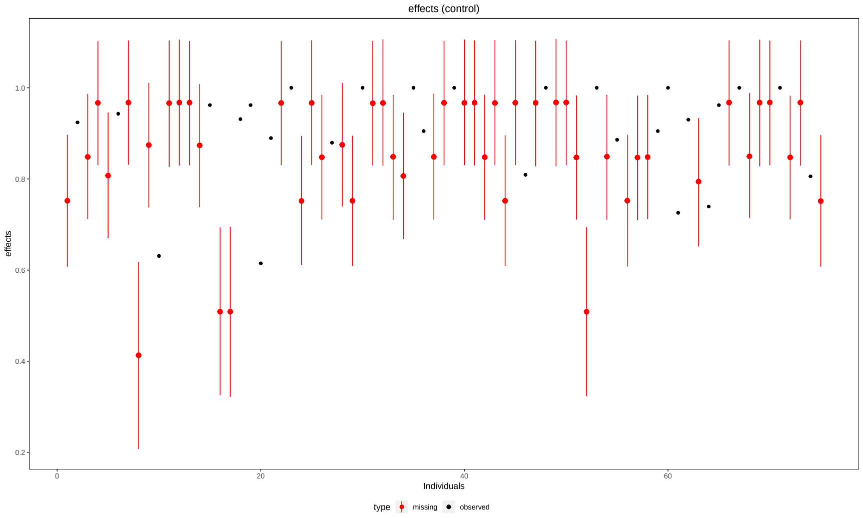
e We can use the function `plot` to display the distribution of the observed and imputed data in both outcome variables and treatment groups. The following command

```
> NN.sel.imp=plot(NN.sel)
```



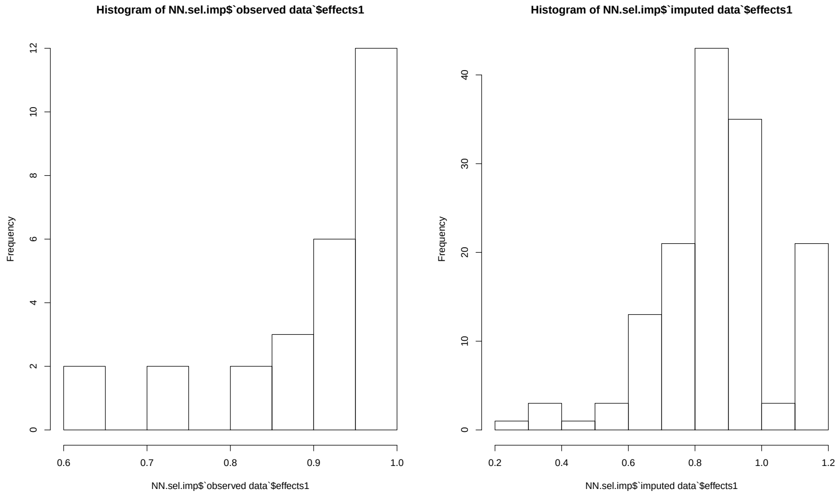
returns the plot of the observed (denoted with black dots) and imputed data, the latter being represented in terms of the posterior means and 95% credible intervals (denoted with red dots and lines). Different graphs are displayed for the QALYs and costs variables in each treatment group. Overall, imputed values for both outcomes exceed the natural range of the variables, i.e. imputed QALYs and costs can be higher then one and lower than zero, respectively. This suggests that Normal distributions can generate implausible imputed values for these variables and therefore may lead to incorrect inferences. We can also use `plot` to display the imputed values for only one type of outcome in a specific treatment group. For example, the imputed QALYs in the control group can be shown by setting the optional argument `outcome="effects_arm1"` and using the following command

```
> NN.sel.imp=plot(NN.sel, outcome = "effects_arm1")
```



The object `NN.sel.imp` is a list which contains the observed and imputed QALYs in each treatment group. For example, we can use the following commands to compare the histograms of the observed and imputed QALYs in the control group.

```
> par(mfrow=c(1,2))
> hist(NN.sel.imp$`observed data`$effects1)
> hist(NN.sel.imp$`imputed data`$effects1)
```



The two plots show clear differences between the distribution of the observed and imputed data, with the latter that can also exceed the upper boundary of one.

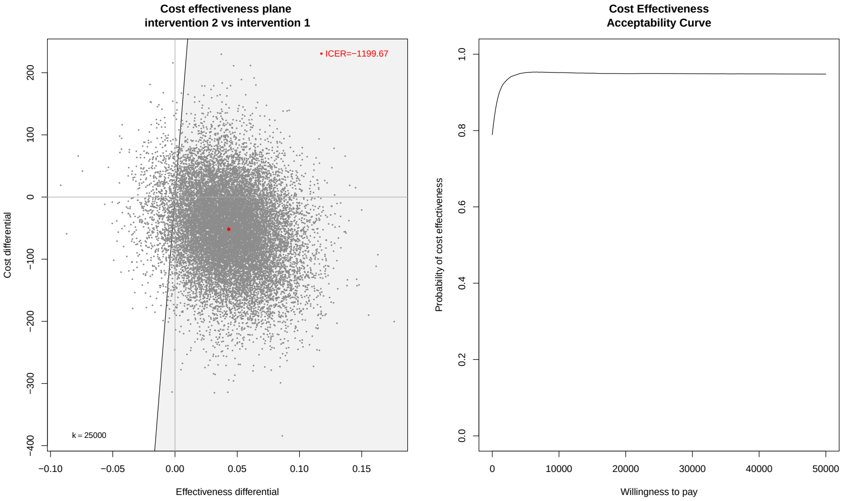
f We can use the **summary** function to summarise the economic results from the model using the following command.

```
> summary(NN.sel)

 Cost-effectiveness analysis summary

 Comparator intervention: intervention 1
 Reference intervention: intervention 2

 Parameter estimates under MAR assumption

 Comparator intervention
               mean     sd      LB      UB
mean.effects   0.874  0.017   0.846   0.901
mean.costs   237.773 51.689 154.097 323.895

 Reference intervention
                 mean     sd      LB      UB
mean.effects.1   0.917  0.022   0.881   0.953
mean.costs.1   185.971 41.381 117.595 252.669

 Incremental results
                  mean     sd      LB     UB
delta.effects     0.043  0.028   -0.001  0.089
delta.costs     -51.801 66.163 -161.276 55.136
ICER          -1199.672
```

The mean QALYs and costs on average are higher in the intervention and control group, respectively, as also indicated by the values of the mean differentials `delta.effects` and `delta.costs` between the two groups. This is also reflected in the negative value of the ICER, which indicates that the new intervention dominates the control.

We can load the package **BCEA** and use the functions `ceplane.plot` and `ceac.plot` to display the cost-effectiveness plane and cost-effectiveness acceptability curve based on the results from the model. These two plots can be generated using the following commands.

```
> library(BCEA)
> par(mfrow=c(1,2))
> ceplane.plot(NN.sel$cea)
> ceac.plot(NN.sel$cea)
```



Both graphs indicate that the new intervention has a high chance of being cost-effective with respect to the control for most values of the acceptance threshold.

g We can modify different types of assumptions of the model to assess the impact of alternative specifications on the final results. Alternative models can be fitted by changing the value of the arguments in the `selection` function. For example, we can include the covariate age in both missingness models (`model.me` and `model.mc`) to specify a MAR mechanism. Finally, we use Beta and Gamma distributions to model the QALYs and cost variables, respectively. Since these distributions are not defined when $e_i = 1$ and $c_i = 0$, we subtract/add a small constant to the two variables to avoid the boundary values. The commands used to modify the data and fit the models are the following.

```
> set.seed(123)
> MenSS.star=MenSS
> MenSS.star$e=MenSS$e-0.05
> MenSS.star$c=MenSS$c+0.05
> BG.sel=selection(data = MenSS.star, model.eff = e ~ u.0, model.cost = c ~ e,
+  model.me = me ~ age, model.mc = mc ~ age, type = "MAR", n.chains = 2,
+  n.iter = 10000, n.burnin = 1000, dist_e = "beta", dist_c = "gamma",
+  save_model = TRUE)
```

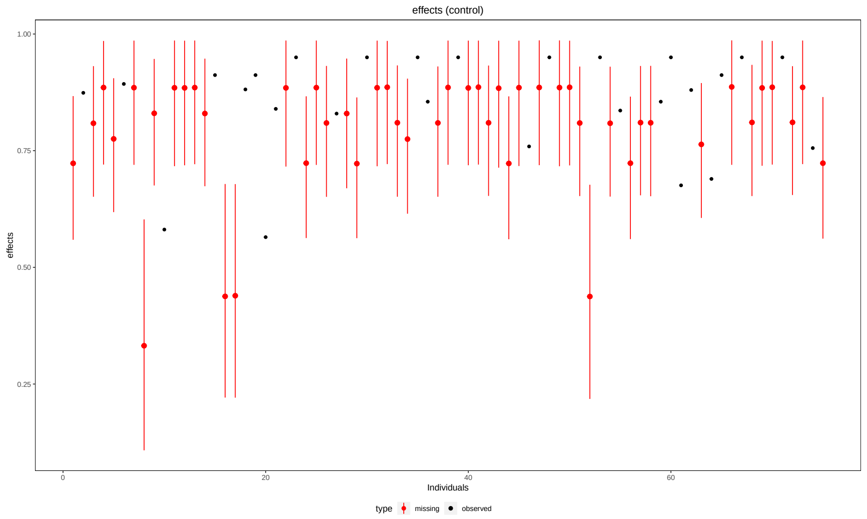
The first two lines are used to set the random seed and create a second version of the dataset (`MenSS.star`). The third and fourth lines rescale the QALYs and costs in this new dataset by subtracting and adding a small constant, respectively. The fifth line re-fits the model under the specification described above.

A quick inspection of the posterior results for the mean QALYs and costs under the new model can be obtained using the following command.

```
> BG.sel$model_output$summary[grep("mu",rownames(BG.sel$model_output$summary)),
+  c("mean","sd","2.5%","97.5%","Rhat","n.eff")]
##            mean      sd    2.5%   97.5%  Rhat n.eff
## mu_c[1] 247.408  88.202 129.871 467.857 1.017   130
## mu_c[2] 292.306 175.696 112.600 815.851 1.001  6100
## mu_e[1]   0.829   0.017   0.792   0.860 1.001 18000
## mu_e[2]   0.862   0.020   0.818   0.897 1.001  4300
```

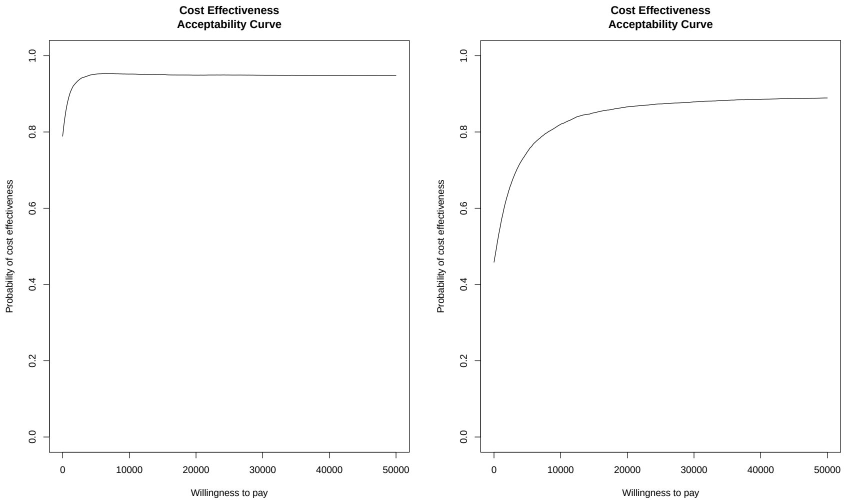
Next, we use the function `plot` to show the imputed QALYs in the control group for the Beta-Gamma model.

```
> BG.sel.imp=plot(BG.sel, outcome = "effects_arm1")
```



Imputations under the Beta-Gamma model seem more reasonable compared with those from the bivariate Normal model since all QALYs are defined within the range of the observed data. Finally, we visually compare the economic results from the bivariate Normal and Beta-Gamma models. Cost-effectiveness acceptability curves can then be displayed using the following commands.

```
> par(mfrow=c(1,2))
> ceac.plot(NN.sel$cea)
> ceac.plot(BG.sel$cea)
```



The results from the Beta-Gamma model (right graph) show a probability of cost-effectivness of the new intervention wich is shifted downwards compared with that from the bivariate Normal model (left graph). Thus, under MAR, the bivariate Normal model seems to overestimate the true cost-effectiveness of the new intervention compared with the Beta-Gamma model. This is also suggested when assessing the fit of the two models using the DIC, which can be obtained through the function `pic`.

```
> pic(NN.sel)[3]
## $dic
## [1] 921.1765
> pic(BG.sel)[3]
## $dic
## [1] 789.5066
```

The better fit of the Beta-Gamma model is indicated by the fact that the DIC associated with this model is lower compared with that from the bivariate Normal model.

Exercise 3: Pattern mixture models

a Follow the script and fit the bivariate Normal model using the `pattern` function

```
> set.seed(123)
> NN.pat=pattern(data = MenSS, model.eff = e~u.0, model.cost = c~e, type = "MAR",
+  n.chains = 2, n.iter = 10000, n.burnin = 1000, dist_e = "norm", dist_c = "norm",
+  Delta_e = 0, Delta_c = 0, save_model = TRUE)
```

b The `JAGS` model generated in the file `pattern.txt` by the commands above is similar to the one fitted with the `selection` function, but is now specified within each missingness pattern $\boldsymbol{r}$. Since, in the MenSS trial, the individual QALYs and costs can be either fully-observed ($r = (1,1)$) or completely missing ($r = (0,0)$), the model can only be fitted within these two missingness patterns. The model is specified as follows.

First, a marginal Normal distribution for the effectiveness is assumed within each pattern $\boldsymbol{r} = [(1,1),(0,0)]$

$$e_i \sim \text{Normal}(\phi^{\boldsymbol{r}}_{iet}, \sigma^{\boldsymbol{r}}_{et}),$$

and the usual linear regression is used to control for the baseline utilities

$$\phi^{\boldsymbol{r}}_{iet} = \alpha^{\boldsymbol{r}}_{0t} + \alpha^{\boldsymbol{r}}_{1t} u_{i0t}.$$

The marginal mean of the QALYs in each pattern $\mu^{\boldsymbol{r}}_{et}$ is then calculated by replacing the baseline utilities with their mean values

$$\mu^{\boldsymbol{r}}_{et} = \alpha^{\boldsymbol{r}}_{0t} + \alpha^{\boldsymbol{r}}_{1t} \bar{u}_{0t}.$$

Next, a similar approach is used to model the costs

$$c_i \sim \text{Normal}(\phi^{\boldsymbol{r}}_{ict}, \sigma^{\boldsymbol{r}}_{ct}),$$

where the mean is specified as the linear predictor

$$\phi^{\boldsymbol{r}}_{ict} = \beta^{\boldsymbol{r}}_{0t} + \beta^{\boldsymbol{r}}_{1t}(e_i - \mu^{\boldsymbol{r}}_{et}).$$

The marginal mean costs can be identified with the intercept term, i.e. $\mu^{\boldsymbol{r}}_{ct} = \beta^{\boldsymbol{r}}_{0t}$. The model for the missingness patterns is specified using a Multinomial distribution,

$$\boldsymbol{r}_i \sim \text{Multinomial}(\boldsymbol{\lambda}^{\boldsymbol{r}}_t),$$

where $\boldsymbol{r} \in \{(1,1),(0,0)\}$, while $\boldsymbol{\lambda}^{\boldsymbol{r}}_t$ denotes the pattern probabilities conditional on the treatment assignment t. The `JAGS` code is replicated for the two treatment groups and by default it specifies vague prior distributions on all parameters that index the distribtuion of the observed QALYs and costs (i.e. in the pattern $r = (1,1)$) and for $\boldsymbol{\lambda}^{\boldsymbol{r}}_t$.

- $\boldsymbol{\alpha} = (\alpha^{r=(1,1)}_{0t}, \alpha^{r=(1,1)}_{1t}) \overset{iid}{\sim} \text{Normal}(0, 0.0000001)$
- $\boldsymbol{\beta} = (\beta^{r=(1,1)}_{0t}, \beta^{r=(1,1)}_{1t}) \overset{iid}{\sim} \text{Normal}(0, 0.0000001)$
- $\log\boldsymbol{\sigma}^{r=(1,1)} = (\log\sigma^{r=(1,1)}_{et}, \log\sigma^{r=(1,1)}_{ct}) \overset{iid}{\sim} \text{Uniform}(-5, 10)$
- $\boldsymbol{\lambda}^{\boldsymbol{r}}_t \sim \text{Dirichlet}(1, \ldots, 1)$

Next, identifying restrictions and sensitivity parameters are used to identify the distribution of the missing data. Since the key quantities of interest for the economic analysis are the mean QALYs and costs, the distribution of missingness is identified only up to these parameters. More specifically, the marginal mean outcome parameters in $r = (0,0)$ are identified using the corresponding means estimated from the completers $r = (1,1)$ (complete case restriction) and some sensitivity parameters $\boldsymbol{\Delta}_t = (\Delta_{et}, \Delta_{ct})$:

$$\mu^{r=(0,0)}_{et} = \mu^{r=(1,1)}_{et} + \Delta_{et} \quad \text{and} \quad \mu^{r=(0,0)}_{ct} = \mu^{r=(1,1)}_{ct} + \Delta_{ct}.$$

Under MAR, both Δ_{et} and Δ_{ct} are set to 0, while under MNAR Uniform prior distributions are assumed on $\boldsymbol{\Delta}_t$, whose hyperprior values $\boldsymbol{\delta}_t$ must be provided by the user:

$$\Delta_{et} \sim \text{Uniform}(\delta^1_{et}, \delta^2_{et}) \quad \text{and} \quad \Delta_{ct} \sim \text{Uniform}(\delta^1_{ct}, \delta^2_{ct}).$$

Once the mean parameters indexing the distribution of the missing data have been identified, then the overall mean QALYs and costs are calculated as weighted averages of the means across all patterns, i.e. $\mu_{et} = \sum_r \mu^r_{et}\lambda^r_t$ and $\mu_{ct} = \sum_r \mu^r_{ct}\lambda^r_t$.

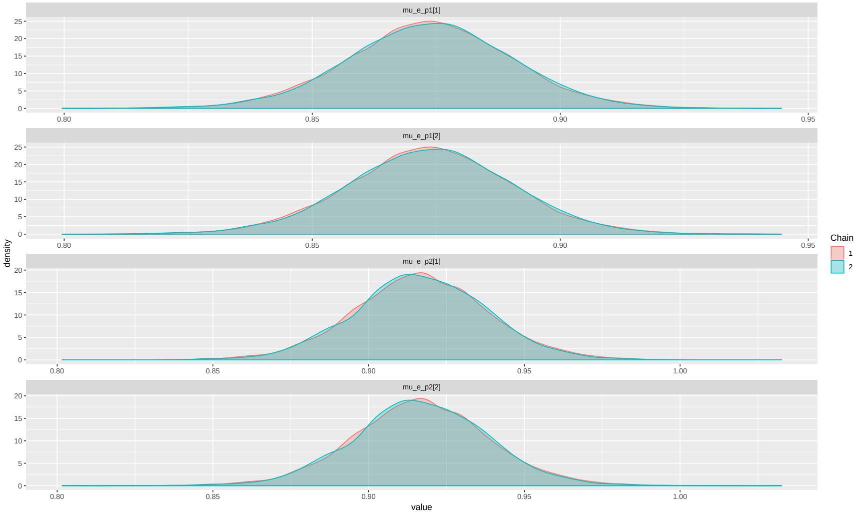
c Selected summary statistics for the mean QALY and cost parameters can be obtained by using the following command

```
> NN.pat$model_output$summary[grep("mu",rownames(NN.pat$model_output$summary)),
+  c("mean","sd","2.5%","97.5%","Rhat","n.eff")]
##                mean     sd    2.5%   97.5%  Rhat n.eff
## mu_c[1]     238.022 51.874 137.526 342.810 1.001 18000
## mu_c[2]     186.803 41.330 104.882 269.245 1.001  6600
## mu_c_p1[1] 238.022 51.874 137.526 342.810 1.001 18000
## mu_c_p1[2] 238.022 51.874 137.526 342.810 1.001 18000
## mu_c_p2[1] 186.803 41.330 104.882 269.245 1.001  6600
## mu_c_p2[2] 186.803 41.330 104.882 269.245 1.001  6600
## mu_e[1]       0.874  0.017   0.840   0.907 1.001 18000
## mu_e[2]       0.917  0.022   0.874   0.960 1.001 18000
## mu_e_p1[1]    0.874  0.017   0.840   0.907 1.001 18000
## mu_e_p1[2]    0.874  0.017   0.840   0.907 1.001 18000
## mu_e_p2[1]    0.917  0.022   0.874   0.960 1.001 18000
## mu_e_p2[2]    0.917  0.022   0.874   0.960 1.001 18000
```

Estimates are reported for both the pattern-specific (`mu_e_p` and `mu_c_p`) and overall means (`mu_e` and `mu_c`). Since, in the MenSS trial, QALYs and costs can be only either completely observed or missing and the model identifies the means in $r = (0, 0)$ using those from $r = (1, 1)$, under MAR, the estimates in the two patterns are the same and also coincide with the overall means μ_{et} and μ_{ct}.

d The usual diagnostic plots for all model parameters can be obtained through the function `diagnostic`. When the model is fitted using the function `pattern` it is also possible to display the diagnostics for the pattern-specifc mean QALYs, mean costs and probabilities using the argument `param="mu.e.p"`, `param="mu.c.p"` and `param="pattern"`, respectively. For example, the density plots for the mean QALYs by missingness pattern and treatment group can be obtained using the following command

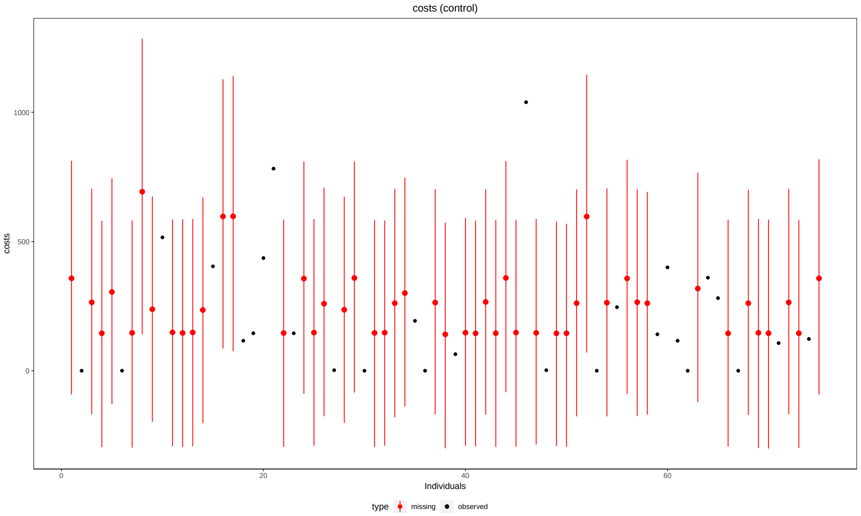
```
> diagnostic(NN.pat, type = "denplot", param = "mu.e.p")
```



where `mu_e_p1` and `mu_e_p2` indicate the mean QALYs in $r = (1, 1)$ and $r = (0, 0)$, while the indices `[1]` and `[2]` are associated with the control and intervention group, respectively.

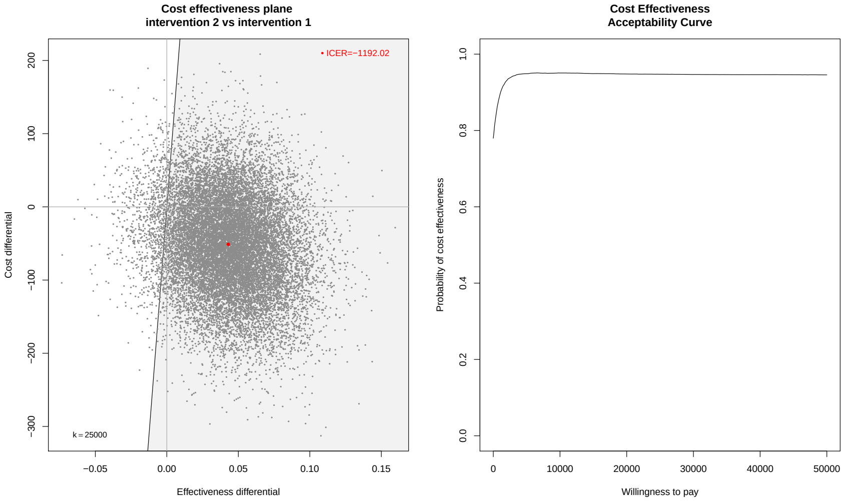
The distribution of the missing data can be inspected using the usual command `plot`. For example, the plot of the observed and imputed costs in the control group can be obtained by typing

```
> NN.pat.imp=plot(NN.pat, outcome = "costs_arm1")
```



e Using the functions in the package **BCEA** we obtain the cost-effectiveness plane and cost-effectiveness acceptability curve based on the results from the model.

```
> par(mfrow=c(1,2))
> ceplane.plot(NN.pat$cea)
> ceac.plot(NN.pat$cea)
```



The cost-effectiveness results under MAR are almost identical with respect to those obtained from using the `selection` function since the parameters in the model are estimated from the same observed data and the only patterns in the dataset are $r = (1, 1)$ and $r = (0, 0)$.

f We now change the specification of the model and fit it under a MNAR assumption for the QALYs. We first define the hyperprior values of the prior distributions of the sensitivity parameters Δ_{et} using the following commands

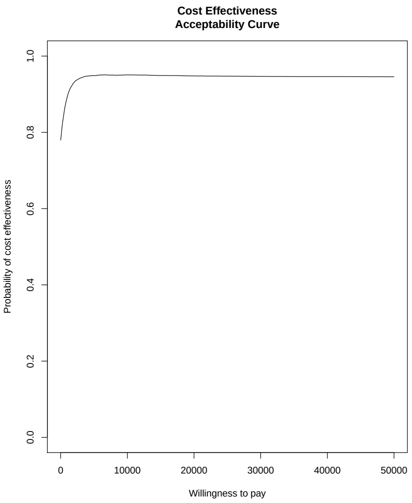
```
> prior.Delta.e=matrix(NA, nrow = 2, ncol = 2)
> prior.Delta.e[,1]=c(-0.2, -0.2)
> prior.Delta.e[,2]=c(-0.1, -0.1)
```

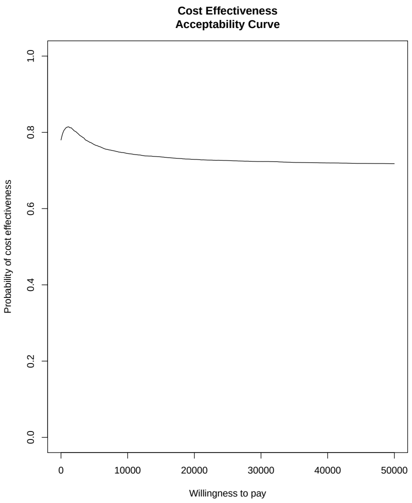
The first line creates the 2×2 matrix `prior.Delta.e` which is then filled-in by the commands in the last two lines. Specifically, the lower and upper bounds for the distributions of Δ_{et} (-0.2 and -0.1) are provided for the control (first row) and intervention (second row) group. Next, the object `prior.Delta.e` is passed to the argument `Delta.e` in the `pattern` function and the argument `type` is set to `"MNAR"`. The model can be fitted by typing

```
> set.seed(123)
> NN.pat.mnar=pattern(data = MenSS, model.eff = e~u.0, model.cost = c~e,
+  type = "MNAR", n.chains = 2, n.iter = 10000, n.burnin = 1000, dist_e = "norm",
+  dist_c = "norm", Delta_e = prior.Delta.e, Delta_c = 0, save_model = TRUE)
```

A summary comparison of the cost-effectiveness results between the pattern mixture model fitted under MAR and MNAR can be obtained using the following commands

```
> par(mfrow=c(1,2))
> ceac.plot(NN.pat$cea)
> ceac.plot(NN.pat.mnar$cea)
```





The cost-effectiveness probability is considerably lower for the model fitted under MNAR (right graph) compared with MAR (left graph). This suggests that the results under MAR are not robust to the MNAR departure explored and can potentially lead to incorrect conclusions.

Exercise 4: Hurdle models

a Follow the script and fit the bivariate Normal model using the `hurdle` function

```
> set.seed(123)
> NN.hur=hurdle(data = MenSS, model.eff = e ~ u.0, model.cost = c ~ e,
+  model.se = se ~ 1, model.sc = sc ~ 1, type = "SCAR", se = 1, sc = 0,
+  n.chains = 2, n.iter = 10000, n.burnin = 1000,
+  dist_e = "norm", dist_c = "norm", save_model = TRUE)
```

b The `JAGS` model generated in the file `hurdle.txt` by the commands above is similar to the one fitted with the `selection` function, but replaces the missingness models with the models for the structural value indicators for the QALY and cost data (d_{iet}, d_{ict}), which are indicated by the arguments `model.se` and `model.sc`, respectively. The choice of the observed values that should be considered as "structural" by the model is made through the arguments `se` and `sc`; in case the structural values are only observed for one outcome, for example the effects, it is possible to set `sc=NULL` to fit the hurdle model only to the effect variables. Since in the MenSS trial both unit QALYs and zero costs are observed, a hurdle model is specified to handle both types of structural values. The model is specified as follows.

The `JAGS` model uses a different sampling distribution for both the effects and costs, depending on the observed values of the indicators d_{ie} and d_{ic}, respectively. Thus, the models for the effects and costs can be represented as mixture models, each formed by two components: $e_i < 1$ and $e_i = 1$ for the effects, and $c_i > 0$ and $c_i = 0$ for the costs. For example, the marginal model for the effects can be represented as

$$e_i \sim \text{Normal}(\phi_{iet}^{d_{ie}}, \sigma_{et}^{d_{ie}}).$$

When $d_{ie} = 1$ (i.e. $e_i = 1$) a degenerate distribution at a point mass at 1 is fitted to the data, while when $d_{ie} = 0$ (i.e. $e_i < 1$) a Normal distribution is fitted to the data. The usual linear regression is used to control for the baseline utilities

$$\phi^{d_{ie}}_{iet} = \alpha^{d_{ie}}_{0t} + \alpha^{d_{ie}}_{1t} u_{i0},$$

where $\alpha^0_{0t} = 1$ and $\alpha^0_{1t} = 0$ for those individuals with $e_i = 1$. The marginal means of the QALYs in the two mixture components are then calculated as

$$\mu^0_{et} = \alpha^0_{0t} + \alpha^0_{1t}\bar{u}_{0t} \quad \text{and} \quad \mu^1_{et} = \alpha^1_{0t}.$$

A similar approach is used for the model of the costs

$$c_{it} \sim \text{Normal}(\phi^{d_{ic}}_{ict}, \sigma^{d_{ic}}_{ct}).$$

When $d_{ic} = 1$ (i.e. $c_i = 0$) a degenerate distribution at a point mass at 0 is fitted to the data, while when $d_{ic} = 0$ (i.e. $c_i > 0$) a Normal distribution is fitted to the data. The conditional mean cost is specified as the linear predictor

$$\phi^{d_{ic}}_{ict} = \beta^{d_{ic}}_{0t} + \beta^{d_{ic}}_{1t}(e_i - \mu_{et}),$$

where $\beta^0_{0t} = 0$ and $\beta^0_{1t} = 0$ for those individuals with $c_i = 0$. The marginal means of the costs in the two mixture components are then calculated as

$$\mu^0_{ct} = \beta^0_{0t} \quad \text{and} \quad \mu^1_{ct} = \beta^1_{0t}.$$

The models for the structural value indicators is specified using Bernoulli distributions,

$$d_{ie} \sim \text{Bernoulli}(\pi_{et}) \quad \text{and} \quad d_{ic} \sim \text{Bernoulli}(\pi_{ct})$$

where the probability of having a structural value is modelled on the logit scale as

$$\text{logit}(\pi_{et}) = \gamma_{et} \quad \text{and} \quad \text{logit}(\pi_{ct}) = \gamma_{ct}.$$

Since no covariates are included in the models, the marginal mean probabilities are calculated using the inverse logit function as $\bar{\pi}_{et} = \frac{\exp(\gamma_{et})}{1+\exp(\gamma_{et})}$ and $\bar{\pi}_{ct} = \frac{\exp(\gamma_{ct})}{1+\exp(\gamma_{ct})}$. These marginal probabilities are then used as weights to calculate the marginal mean QALYs and costs in each treatment group across the two components of the model:

$$\mu_{et} = \mu^0_{et}(1 - \bar{\pi}_{et}) + \mu^1_{et}\bar{\pi}_{et} \quad \text{and} \quad \mu_{ct} = \mu^0_{ct}(1 - \bar{\pi}_{ct}) + \mu^1_{ct}\bar{\pi}_{ct}.$$

The `JAGS` code is replicated for the two treatment groups and by default it specifies vague prior distributions on the parameters indexing the models of d_{ie} and d_{ic} and those indexing the distribution of $e_i < 1$ and $c_i > 0$:

- $\boldsymbol{\alpha}^0 = (\alpha^0_{0t}, \alpha^{0)}_{1t}) \overset{iid}{\sim} \text{Normal}(0, 0.0000001)$
- $\boldsymbol{\beta}^0 = (\beta^0_{0t}, \beta^0_{1t}) \overset{iid}{\sim} \text{Normal}(0, 0.0000001)$
- $\text{log}\boldsymbol{\sigma}^0 = (\text{log}\sigma^0_{et}, \text{log}\sigma^0_{ct}) \overset{iid}{\sim} \text{Uniform}(-5, 10)$
- $\boldsymbol{\gamma} = (\gamma_{et}, \gamma_{ct}) \overset{iid}{\sim} \text{Logistic}(0, 1)$

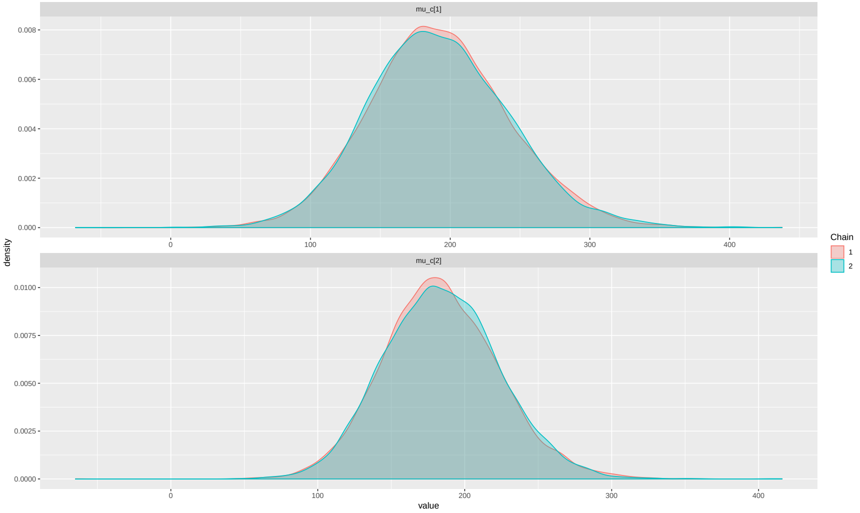
Informative priors are specified on the parameters indexing the distribution of $e_i = 1$ and $c_i = 0$ to induce a variance as close to 0 as possible, i.e. $\boldsymbol{\sigma}^1 = 0.000001$.

c We use the function `print` to show key posterior summaries of the mean QALY and cost parameters

```
> NN.hur$model_output$summary[grep("mu",rownames(NN.hur$model_output$summary)),
+  c("mean","sd","2.5%","97.5%","Rhat","n.eff")]
##             mean     sd    2.5%   97.5%  Rhat n.eff
## mu_c[1] 192.406 50.563  98.677 296.750 1.001 18000
## mu_c[2] 184.264 39.300 110.882 264.820 1.001 13000
## mu_e[1]   0.906  0.020   0.866   0.942 1.001 18000
## mu_e[2]   0.917  0.027   0.858   0.964 1.001 18000
```

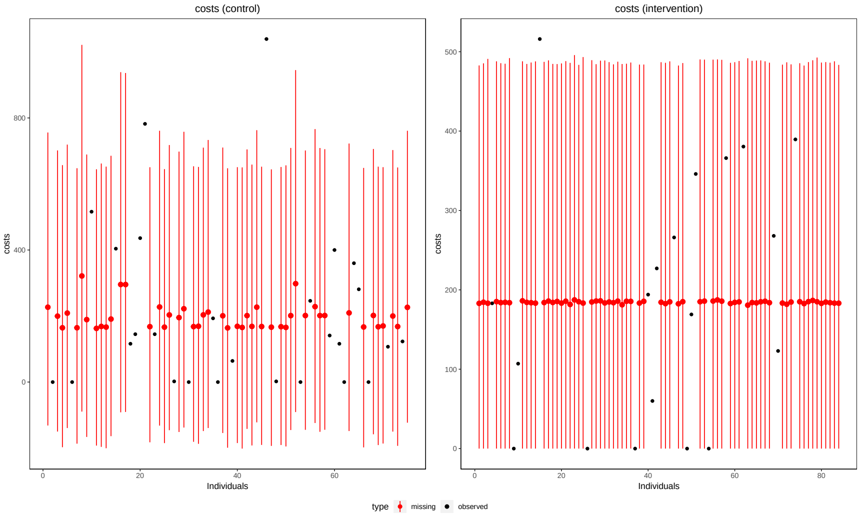
d Density plots for the mean costs in both treatment groups can be visualised using the `diagnostic` function

```
> diagnostic(NN.hur, type = "denplot", param = "mu.c")
```



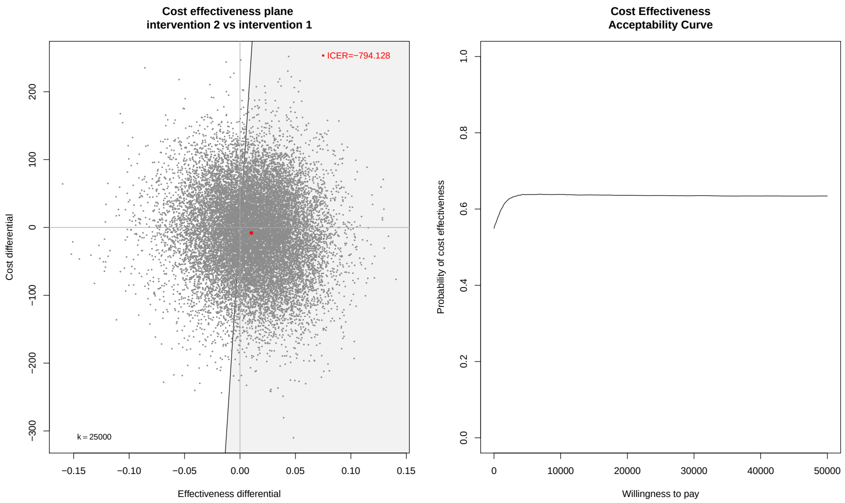
while the function `plot` is used to display the distribution of the imputed and observed cost data in both treatment groups

```
> NN.hur.imp=plot(NN.hur, outcome = "costs")
```



e We display the graphs for the cost-effectiveness plane and cost-effectiveness acceptability curve to summarise the economic results from the model

```
> par(mfrow=c(1,2))
> ceplane.plot(NN.hur$cea)
> ceac.plot(NN.hur$cea)
```



The results from the bivariate Normal model fitted with the function `hurdle` are considerably different with respect to those obtained from using the functions `selection` and `pattern` and suggest that the new intervention is not cost-effective compared with the control.

f We change the model and specify Beta and LogNormal distributions for the QALYs and costs. Since the structural values in both outcomes are handled explicitly, no rescaling is necessary when fitting these distributions to the data using a hurdle approach. We include the baseline utilities as covariates in the model for the structural ones (`model.se`) and specify a structural at random (SAR) mechanism by setting `type="SAR"`. Finally, we fit the model under a MNAR assumption about the structural ones. We assume that all the individuals with a unit baseline utility are also associated with a unit QALYs and construct the corresponding indicator variables using the following commands

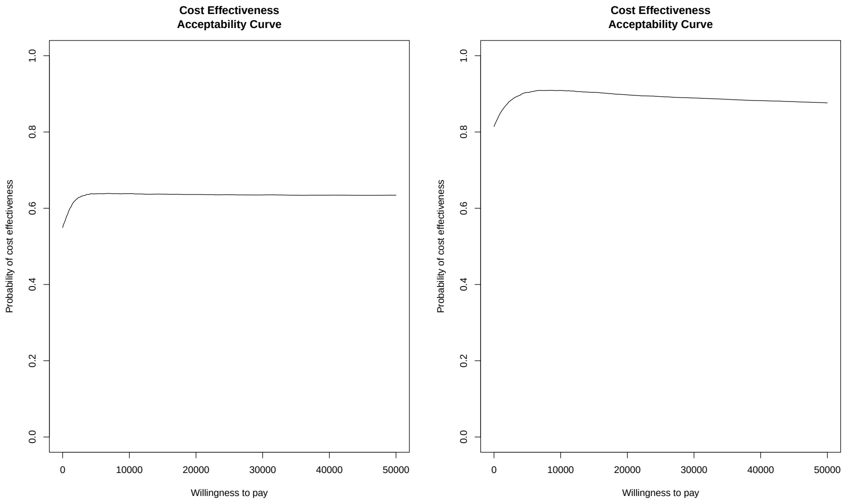
```
> d_e=ifelse(MenSS$e==1, 1, 0)
> d_e[MenSS$u.0==1 & is.na(MenSS$e)]=1
```

We then pass this indicator variable to the `hurdle` function using the optional argument `d_e`. The updated verion of the model can then be fitted using the following commands

```
> set.seed(123)
> BL.hur.mnar=hurdle(data = MenSS, model.eff = e ~ 1, model.cost = c ~ e,
+  model.se = se ~ u.0, model.sc = sc ~ 1, type = "SAR", se = 1, sc = 0,
+  n.chains = 2, n.iter = 10000, n.burnin = 1000, dist_e = "beta",
+  dist_c = "lnorm", save_model = TRUE, d_e=d_e)
```

We compare the economic results between the bivariate Normal and Beta-LogNormal hurdle models in terms of the cost-effectiveness acceptability curves

```
> par(mfrow=c(1,2))
> ceac.plot(NN.hur$cea)
> ceac.plot(BL.hur.mnar$cea)
```



The two graphs lead to different cost-effectiveness conclusions, with the bivariate Normal model (left graph) suggesting a considerably lower probability of cost-effectiveness compared with the Beta-LogNormal model (right graph) for all values of the acceptance threshold considered.