

mice: An approach to sensitivity analysis

Gerko Vink and Stef van Buuren

Vignette 6 of 6

This is the last vignette in the series.

The focus of this document is on sensitivity analysis in the context of missing data. The goal of sensitivity analysis is to study the influence that violations of the missingness assumptions have on the obtained inference.

The Leiden data set

The Leiden data set is a subset of 956 members of a very old (85+) cohort in Leiden. Multiple imputation of this data set has been described in Boshuizen et al (1998), Van Buuren et al (1999) and Van Buuren (2012), chapter 7.

The main question is how blood pressure affects mortality risk in the oldest old. We have reasons to mistrust the MAR assumption in this case. In particular, we worried whether the imputations of blood pressure under MAR would be low enough. The sensitivity analysis explores the effect of artificially lowering the imputed blood pressure by deducting an amount of δ from the values imputed under MAR. In order to preserve the relations between the variables, this needs to be done during the iterations.

Unfortunately we cannot share the Leiden data set with you. But we detail the approach below.

1. Open R and load the packages `mice`, `lattice` and `survival`.

```
set.seed(123)
library("mice")
library("lattice")
library("survival")
```

2. The leiden data set.

```
summary(leiden)
```

```
##      sexe      lftanam      rrsyst      rrdiast
## Min. :0.0000 Min. : 85.48 Min. : 90.0 Min. : 50.00
## 1st Qu.:0.0000 1st Qu.: 87.50 1st Qu.:135.0 1st Qu.: 75.00
## Median :0.0000 Median : 89.07 Median :150.0 Median : 80.00
## Mean :0.2709 Mean : 89.78 Mean :152.9 Mean : 82.78
## 3rd Qu.:1.0000 3rd Qu.: 91.52 3rd Qu.:170.0 3rd Qu.: 90.00
## Max. :1.0000 Max. :103.54 Max. :260.0 Max. :140.00
##
##           NA's :121 NA's :126
##      dwa      survda      alb      chol
## Min. :0.0000 Min. : 2.0 Min. :21.00 Min. : 2.700
## 1st Qu.:0.0000 1st Qu.: 534.8 1st Qu.:39.00 1st Qu.: 4.800
## Median :0.0000 Median :1196.5 Median :41.00 Median : 5.700
## Mean :0.2437 Mean :1195.4 Mean :40.77 Mean : 5.704
## 3rd Qu.:0.0000 3rd Qu.:1889.0 3rd Qu.:43.00 3rd Qu.: 6.400
## Max. :1.0000 Max. :2610.0 Max. :52.00 Max. :10.900
##
##           NA's :229 NA's :232
##      mmse      woon
## Min. : 1.00 Min. :0.000
## 1st Qu.:21.00 1st Qu.:0.000
## Median :26.00 Median :3.000
## Mean :23.67 Mean :1.775
## 3rd Qu.:29.00 3rd Qu.:3.000
## Max. :30.00 Max. :4.000
## NA's :85
```

```
str(leiden)
```

```
## 'data.frame': 956 obs. of 10 variables:
## $ sexe : num 0 0 0 0 0 0 0 1 1 0 ...
## $ lftanam: num 87.8 87.8 89.1 90.3 87.8 ...
## $ rrsyst : num 160 140 155 155 110 120 180 135 130 160 ...
## $ rrdiast: num 100 70 85 90 60 80 75 80 60 90 ...
## $ dwa : num 0 0 0 0 0 0 0 0 0 0 ...
## $ survda : num 1082 27 1604 528 1100 ...
## $ alb : num 41 NA 41 44 37 NA 42 NA 45 46 ...
## $ chol : num 4.4 NA 4.6 3.9 5.3 NA 7.2 NA 5.1 6.5 ...
## $ mmse : num 12 9 25 27 14 NA 28 26 30 14 ...
## $ woon : num 4 3 0 1 0 3 3 0 4 4 ...
```

```
head(leiden)
```

```
## sexe lftanam rrsyst rrdiast dwa survda alb chol mmse woon
## 1 0 87.80 160 100 0 1082 41 4.4 12 4
## 2 0 87.75 140 70 0 27 NA NA 9 3
## 3 0 89.08 155 85 0 1604 41 4.6 25 0
## 4 0 90.29 155 90 0 528 44 3.9 27 1
## 5 0 87.76 110 60 0 1100 37 5.3 14 0
## 6 0 91.39 120 80 0 5 NA NA NA 3
```

```
tail(leiden)
```

```
##  sexe lftanam rrsyst rrdiast dwa survda alb chol mmse woon
## 1229  1  93.85  130   70  0  523 40  5.3  28  0
## 1230  0  92.20  190   90  0 1182 44  5.8  26  3
## 1232  0  95.02  150   80  0  861 35  5.0  28  0
## 1233  0  88.30  120   60  0  129 42  8.6  21  0
## 1235  1  89.02  140   80  0  374 40  5.2  23  0
## 1236  0  85.70  130   65  0 1744 36  7.2  27  3
```

3. Perform a dry run (using `maxit = 0`) in `mice`. List the number of missing values per variable.

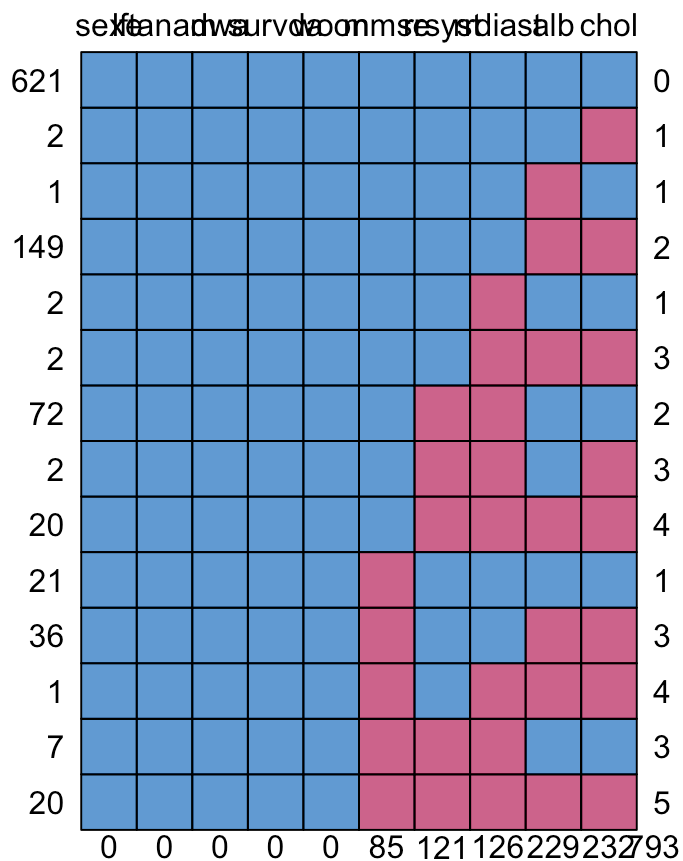
```
ini <- mice(leiden, maxit = 0)
ini$nmis
```

```
##  sexe lftanam rrsyst rrdiast  dwa survda  alb  chol  mmse
##    0    0  121  126    0    0  229  232   85
##  woon
##    0
```

There are 121 missings (NA 's) for `rrsyst` , 126 missings for `rrdiast` , 229 missings for `alb` , 232 missings for `chol` and 85 missing values for `mmse` .

4. Study the missing data pattern in more detail using `md.pattern()` and `fluxplot()` . The interest here focusses on imputing systolic blood pressure (`rrsyst`) and diastolic blood pressure (`rrdiast`).

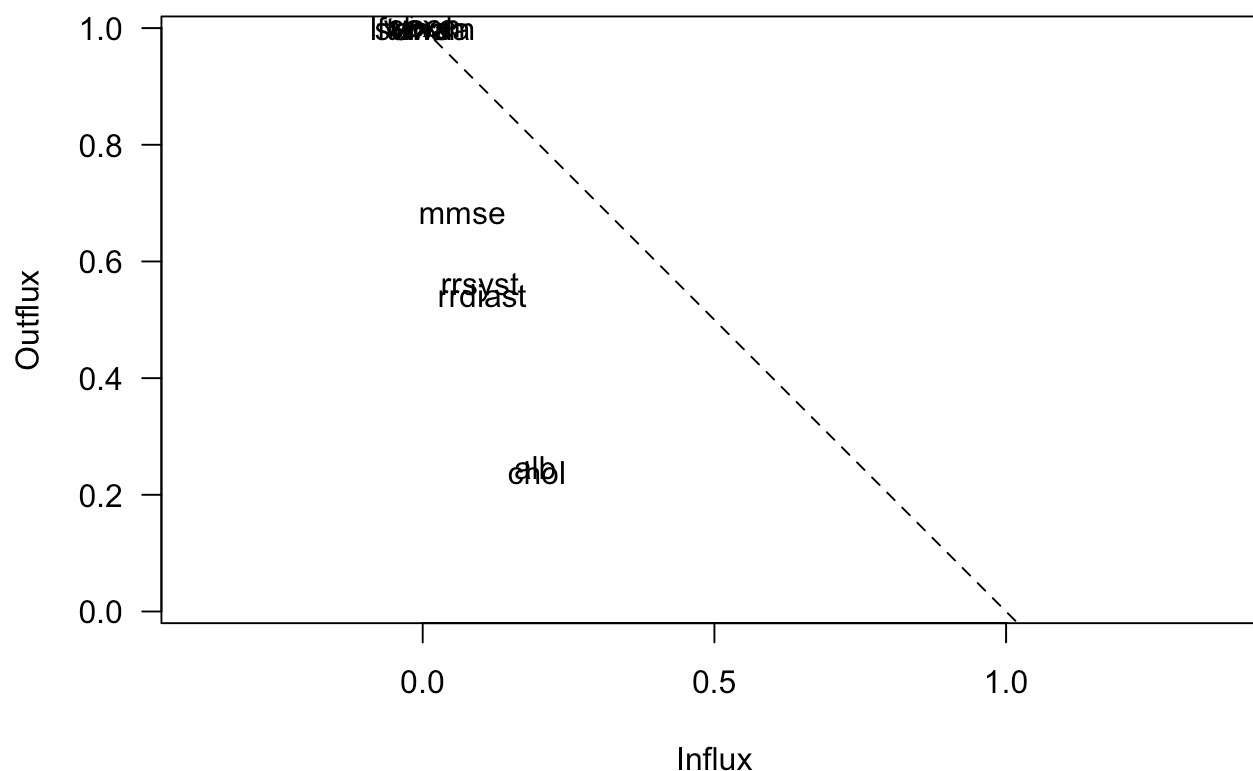
```
md.pattern(leiden)
```



```
##  sexe lftanam dwa survda woon mmse rrsyst rrdiast alb chol
## 621  1      1 1 1 1 1 1 1 1 1 1 0
## 2    1      1 1 1 1 1 1 1 1 1 0 1
## 1    1      1 1 1 1 1 1 1 1 0 1 1
## 149  1      1 1 1 1 1 1 1 1 0 0 2
## 2    1      1 1 1 1 1 1 1 0 1 1 1
## 2    1      1 1 1 1 1 1 1 0 0 0 3
## 72   1      1 1 1 1 1 1 0 0 1 1 2
## 2    1      1 1 1 1 1 1 0 0 1 0 3
## 20   1      1 1 1 1 1 1 0 0 0 0 4
## 21   1      1 1 1 1 1 0 1 1 1 1 1
## 36   1      1 1 1 1 1 0 1 1 0 0 3
## 1    1      1 1 1 1 1 0 1 0 0 0 4
## 7    1      1 1 1 1 1 0 0 0 1 1 3
## 20   1      1 1 1 1 1 0 0 0 0 0 5
##      0      0 0 0 0 0 85 121 126 229 232 793
```

```
fx <- fluxplot(leiden)
```

Influx-outflux pattern for leiden



Variables with higher outflux are (potentially) the more powerful predictors. Variables with higher influx depend stronger on the imputation model. When points are relatively close to the diagonal, it indicates that influx and outflux are balanced.

The variables in the upper left corner have the more complete information, so the number of missing data problems for this group is relatively small. The variables in the middle have an outflux between 0.5 and 0.8, which is small. Missing data problems are thus more severe, but potentially this group could also contain important variables. The lower (bottom) variables have an outflux with 0.5 or lower, so their predictive power is limited. Also, this group has a higher influx, and, thus, depend more highly on the imputation model.

If you'd like this information in tabulated form, you can simply ask

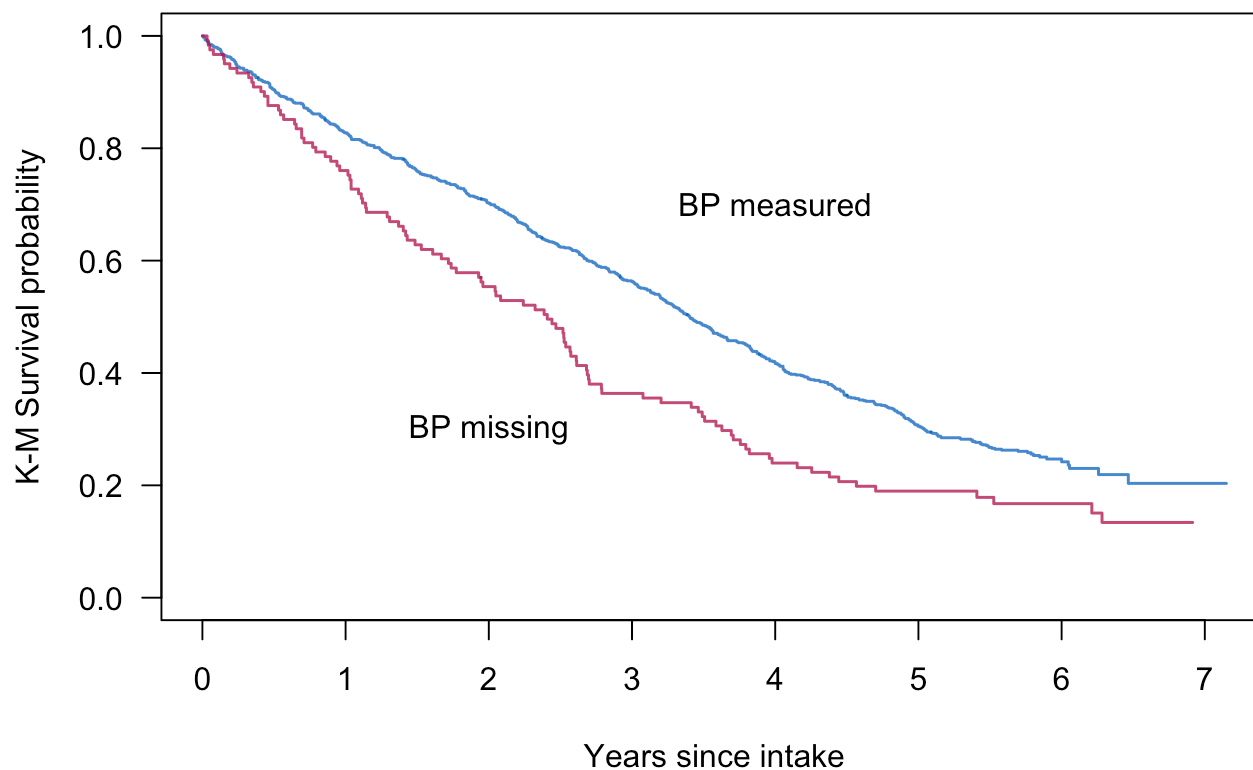
```
fx
```

```
##      pobs  influx outflux  ainb  aout  fico
## sexe  1.000000 0.0000000 1.000000 0.000000 0.09216643 0.3504184
## lftanam 1.000000 0.0000000 1.000000 0.000000 0.09216643 0.3504184
## rrsyst 0.8734310 0.09798107 0.5573770 0.7887971 0.05881570 0.2562874
## rrdiast 0.8682008 0.10231550 0.5422446 0.7910053 0.05756359 0.2518072
## dwa    1.000000 0.0000000 1.000000 0.000000 0.09216643 0.3504184
## survda 1.000000 0.0000000 1.000000 0.000000 0.09216643 0.3504184
## alb    0.7604603 0.19311053 0.2471627 0.8214459 0.02995568 0.1458047
## chol   0.7573222 0.19573400 0.2383354 0.8218391 0.02900552 0.1422652
## mmse   0.9110879 0.06798221 0.6796974 0.7790850 0.06875877 0.2870264
## woon   1.000000 0.0000000 1.000000 0.000000 0.09216643 0.3504184
```

5. The cases with and without blood pressure observed have very different survival rates. Show this.

We can see this easily from the Kaplan-Meier plot.

```
km <- survfit(Surv(survda/365, 1-dwa) ~ is.na(rrsyst), data = leiden)
plot(km,
     lty = 1,
     lwd = 1.5,
     xlab = "Years since intake",
     ylab = "K-M Survival probability", las=1,
     col = c(mdc(4), mdc(5)),
     mark.time = FALSE)
text(4, 0.7, "BP measured")
text(2, 0.3, "BP missing")
```



In the next steps we are going to impute `rrsyst` and `rriast` under two scenarios: MAR and MNAR. We will use the delta adjustment technique described in paragraph 7.2.3 in Van Buuren (2012)

6. Create a δ vector that represent the following adjustment values for mmHg: 0 for MAR, and -5, -10, -15, and -20 for MNAR.

```
delta <- c(0, -5, -10, -15, -20)
```

The recipe for creating MNAR imputations for $\delta \neq 0$ uses the post-processing facility of mice. This allows to change the imputations on the fly by deducting a value of δ from the values just imputed.

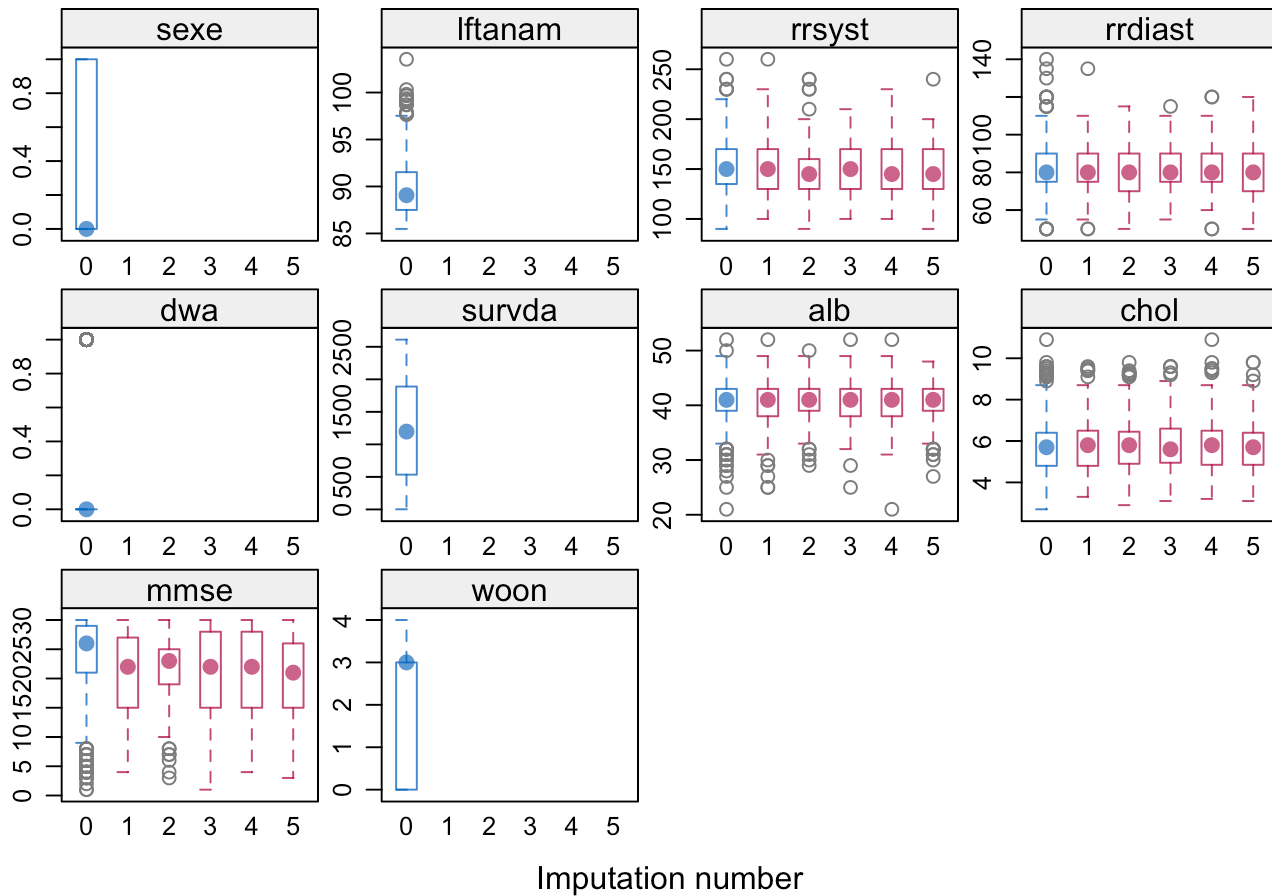
7. Impute the leiden data using the delta adjustment technique. We only have to deduct from `rrsyst`, because `rrdiast` will adapt to the changed `rrsyst` when it is imputed using `rrsyst` as predictor. Store the five imputed scenarios (adjustment) in a list called `imp.all`.

```
imp.all <- vector("list", length(delta))
post <- ini$post
for (i in 1:length(delta)){
  d <- delta[i]
  cmd <- paste("imp[[i]][,i] <- imp[[i]][,i] +", d)
  post["rrsyst"] <- cmd
  imp <- mice(leiden, post = post, maxit = 5, seed = i, print = FALSE)
  imp.all[[i]] <- imp
}
```

8. Inspect the imputations. Compare the imputations for blood pressure under the most extreme scenarios with a box-and-whiskers plot. Is this as expected?

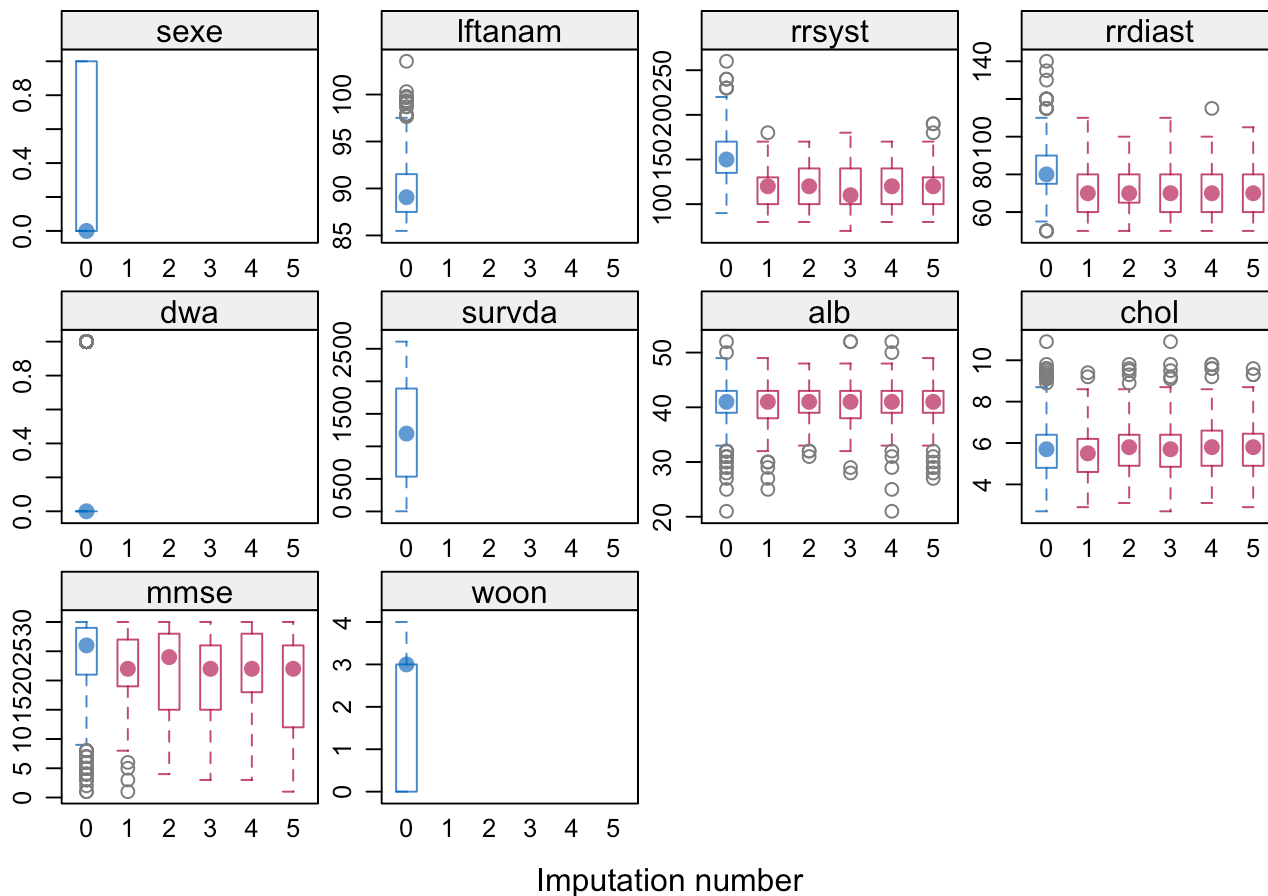
For the scenario where $\delta = 0$ we can plot the first object from the list. This object is the `mids`-object that considers imputations under no adjustment.

```
bwplot(imp.all[[1]])
```



For the scenario where $\delta = -20$ we can plot the fifth object from the list. This object is the `mids`-object that considers imputations under the largest adjustment.

```
bwplot(imp.all[[5]])
```

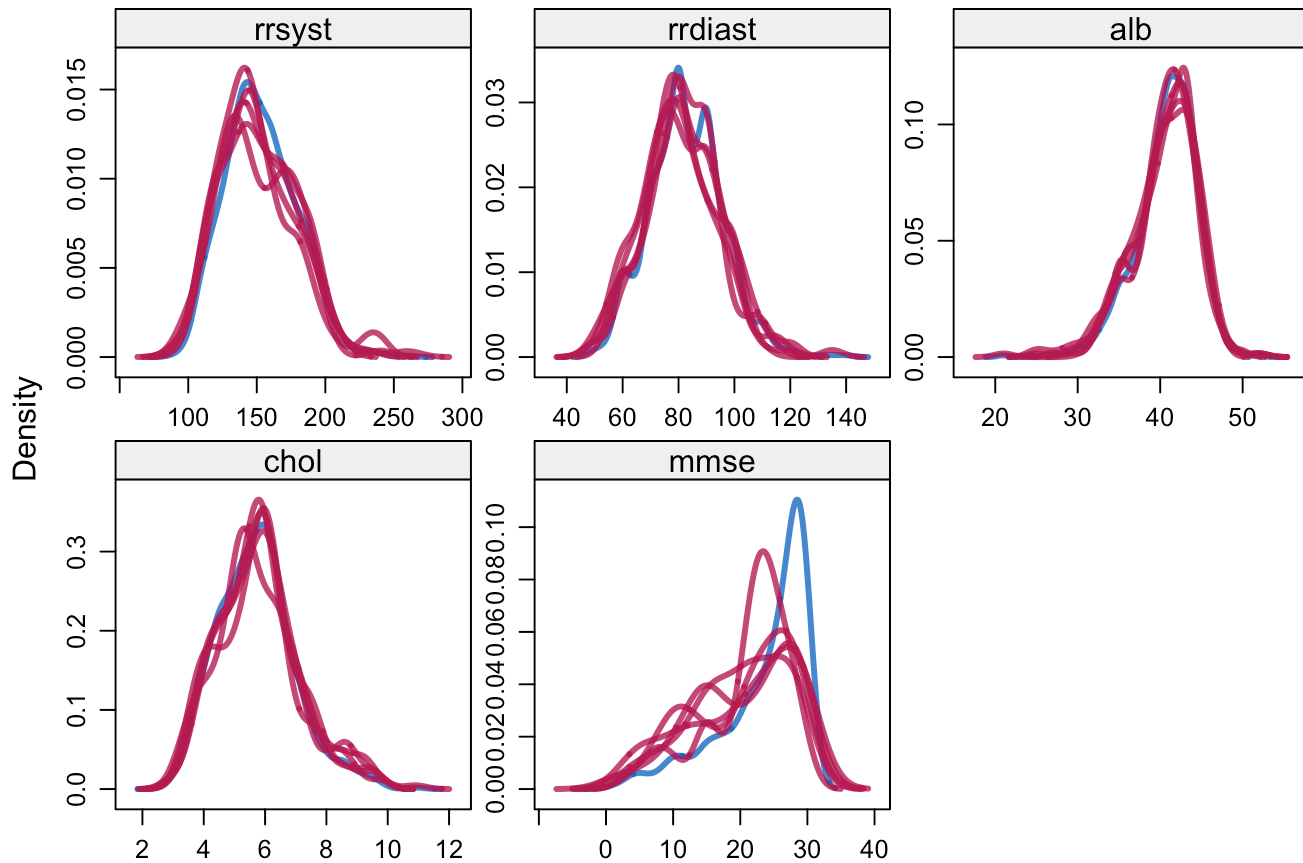



We can clearly see that the adjustment has an effect on the imputations for `rrsyst` and, thus, on those for `rrdiast`.

9. Use the density plot for another inspection.

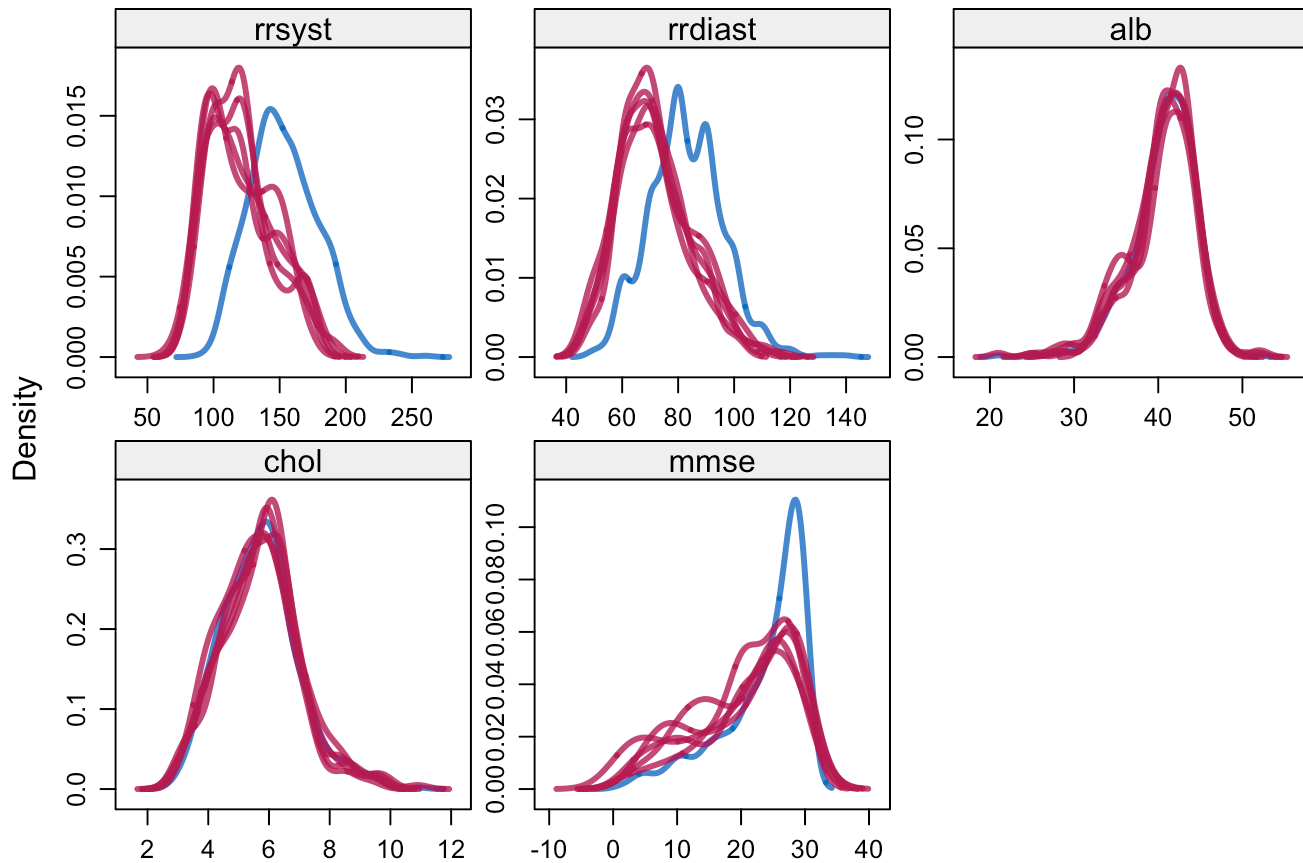
For the scenario where $\delta = 0$ we can plot the first object from the list. This object is the `mids`-object that considers imputations under no adjustment.

```
densityplot(imp.all[[1]], lwd = 3)
```



For the scenario where $\delta = -20$ we can plot the fifth object from the list. This object is the `mids`-object that considers imputations under the largest adjustment.

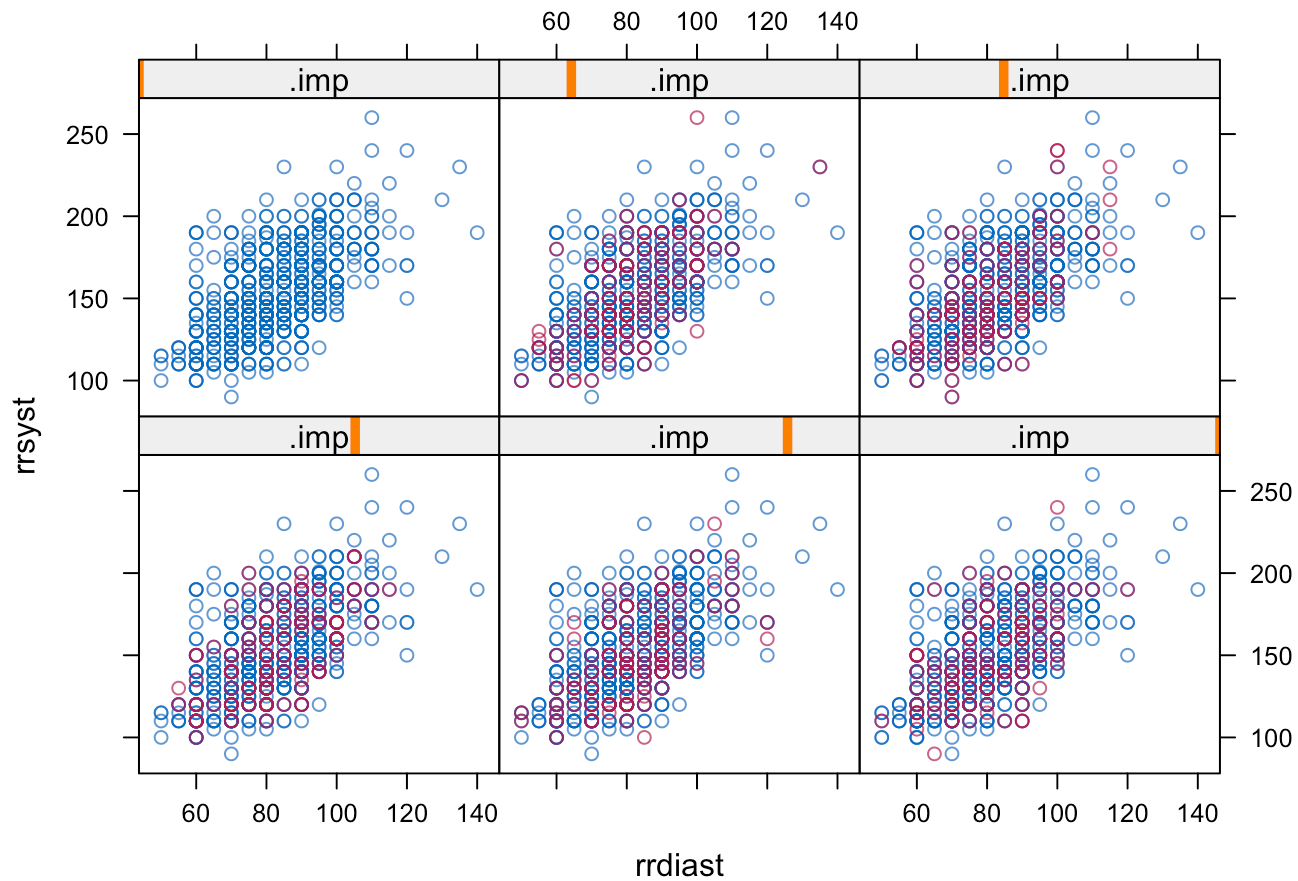
```
densityplot(imp.all[[5]], lwd = 3)
```



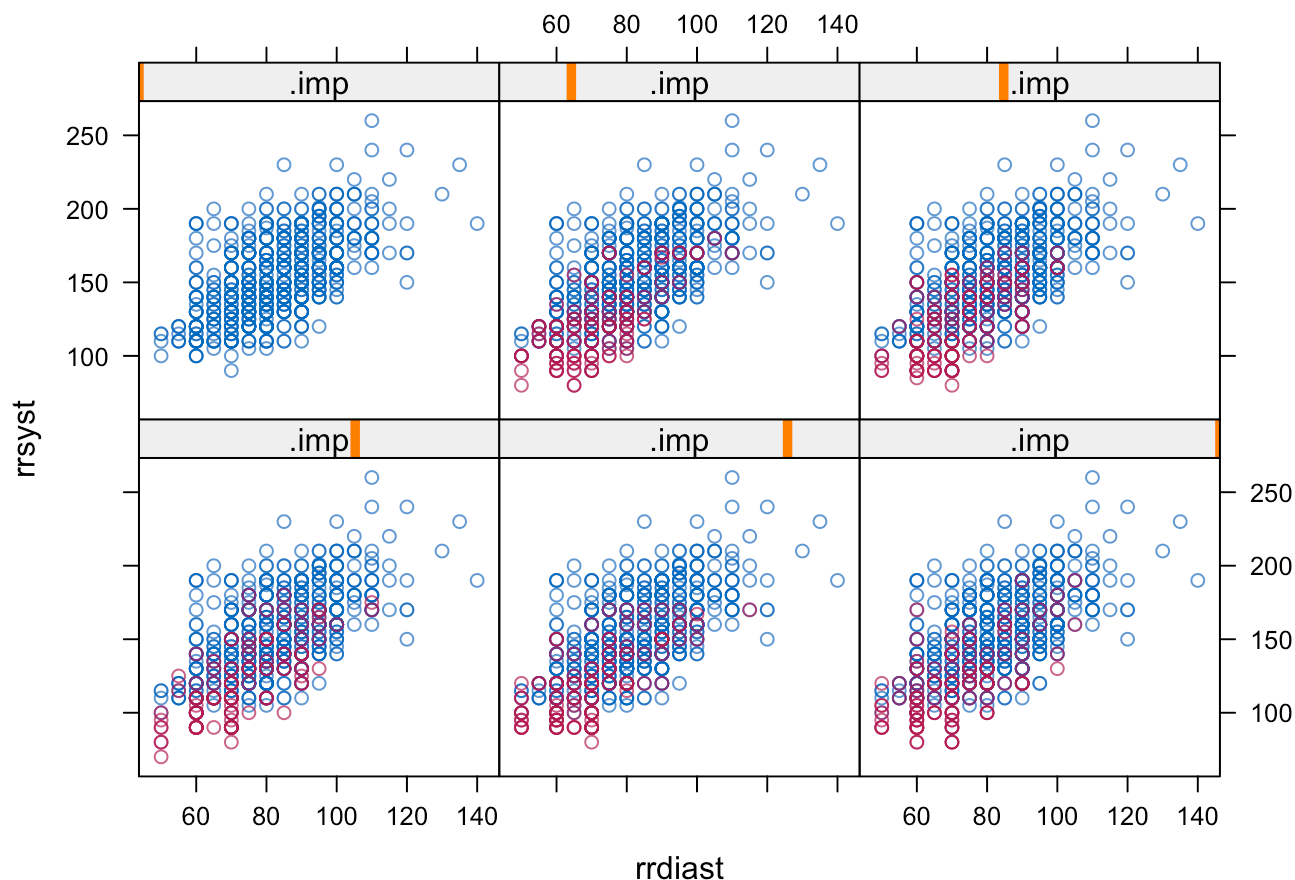
We can once more clearly see that the adjustment has an effect on the imputations for `rrsyst` and, thus, on those for `rrdiast`.

10. Also create a scatter plot of `rrsyst` and `rrdiast` by imputation number and missingness.

```
xyplot(imp.all[[1]], rrsyst ~ rrdiast | .imp)
```



```
xyplot(imp.all[[5]], rrsyst ~ rrdiast | .imp)
```



The scatter plot comparison between `rrsyst` and `rrdiast` shows us that the adjustment has an effect on the imputations and that the imputations are lower for the situation where $\delta = -20$.

We are now going to perform a complete-data analysis. This involves several steps:

1. Create two categorical variables `sbpgp` and `agegp` that divide the observations into groups based on, respectively, systolic blood pressure and age.
2. Calculate whether person died or not.
3. Fit a Cox proportional hazards model to estimate the relative mortality risk corrected for sex and age group.

In order to automate this step we should create an expression object that performs these steps for us. The following object does so:

```
cda <- expression(
  sbpgp <- cut(rrsyst, breaks = c(50, 124, 144, 164, 184, 200, 500)),
  agegp <- cut(lftanam, breaks = c(85, 90, 95, 110)),
  dead <- 1 - dwa,
  coxph(Surv(survda, dead) ~ C(sbpgp, contr.treatment(6, base = 3)) + strata(sexe, agegp))
)
```

See Van Buuren (2012, pp.186) for more information.

11. Create five fit objects that run the expression `cda` on the five imputed adjustment scenarios. Use function `with()`.

```
fit1 <- with(imp.all[[1]], cda)
fit2 <- with(imp.all[[2]], cda)
fit3 <- with(imp.all[[3]], cda)
fit4 <- with(imp.all[[4]], cda)
fit5 <- with(imp.all[[5]], cda)
```

Each fit object contains the five imputed Cox proportional hazards models for the adjustment scenario at hand. For example, the $\delta = -10$ scenario is contained in `fit3`.

```
fit3
```

```
## call :
## with.mids(data = imp.all[[3]], expr = cda)
##
## call1 :
## mice(data = leiden, post = post, maxit = 5, printFlag = FALSE,
##   seed = i)
##
## nmis :
##   sexe lftanam rrsyst rrdiast   dwa survda   alb   chol   mmse
##     0     0   121   126     0     0   229   232    85
##   woon
##     0
##
## analyses :
## [[1]]
## Call:
## coxph(formula = Surv(survda, dead) ~ C(sbpgrp, contr.treatment(6,
##   base = 3)) + strata(sexe, agegp))
##
##               coef exp(coef) se(coef)      z
## C(sbpgrp, contr.treatment(6, base = 3))1 0.49844 1.64616 0.11663 4.274
## C(sbpgrp, contr.treatment(6, base = 3))2 0.34497 1.41194 0.10339 3.337
## C(sbpgrp, contr.treatment(6, base = 3))4 0.07444 1.07728 0.11748 0.634
## C(sbpgrp, contr.treatment(6, base = 3))5 0.09172 1.09606 0.14593 0.629
## C(sbpgrp, contr.treatment(6, base = 3))6 -0.16959 0.84401 0.28744 -0.590
##
##               p
## C(sbpgrp, contr.treatment(6, base = 3))1 1.92e-05
## C(sbpgrp, contr.treatment(6, base = 3))2 0.000848
## C(sbpgrp, contr.treatment(6, base = 3))4 0.526339
## C(sbpgrp, contr.treatment(6, base = 3))5 0.529659
## C(sbpgrp, contr.treatment(6, base = 3))6 0.555183
##
## Likelihood ratio test=25.95 on 5 df, p=9.118e-05
## n= 956, number of events= 723
##
## [[2]]
## Call:
## coxph(formula = Surv(survda, dead) ~ C(sbpgrp, contr.treatment(6,
##   base = 3)) + strata(sexe, agegp))
##
##               coef exp(coef) se(coef)      z
## C(sbpgrp, contr.treatment(6, base = 3))1 0.57702 1.78073 0.11727 4.921
## C(sbpgrp, contr.treatment(6, base = 3))2 0.36627 1.44235 0.10322 3.549
## C(sbpgrp, contr.treatment(6, base = 3))4 0.09947 1.10458 0.12041 0.826
## C(sbpgrp, contr.treatment(6, base = 3))5 0.10371 1.10928 0.14784 0.702
## C(sbpgrp, contr.treatment(6, base = 3))6 -0.07801 0.92495 0.27835 -0.280
##
##               p
## C(sbpgrp, contr.treatment(6, base = 3))1 8.63e-07
## C(sbpgrp, contr.treatment(6, base = 3))2 0.000387
## C(sbpgrp, contr.treatment(6, base = 3))4 0.408747
## C(sbpgrp, contr.treatment(6, base = 3))5 0.482988
## C(sbpgrp, contr.treatment(6, base = 3))6 0.779266
```

```
##
## Likelihood ratio test=31.33 on 5 df, p=8.045e-06
## n= 956, number of events= 723
##
## [[3]]
## Call:
## coxph(formula = Surv(survda, dead) ~ C(sbpgrp, contr.treatment(6,
##   base = 3)) + strata(sexe, agegp))
##
##               coef exp(coef) se(coef)      z
## C(sbpgrp, contr.treatment(6, base = 3))1 0.50643  1.65936  0.11836  4.279
## C(sbpgrp, contr.treatment(6, base = 3))2 0.34137  1.40687  0.10269  3.324
## C(sbpgrp, contr.treatment(6, base = 3))4 0.07230  1.07497  0.11737  0.616
## C(sbpgrp, contr.treatment(6, base = 3))5 0.08921  1.09331  0.14743  0.605
## C(sbpgrp, contr.treatment(6, base = 3))6 -0.22193  0.80097  0.28780 -0.771
##
##               p
## C(sbpgrp, contr.treatment(6, base = 3))1 1.88e-05
## C(sbpgrp, contr.treatment(6, base = 3))2 0.000887
## C(sbpgrp, contr.treatment(6, base = 3))4 0.537906
## C(sbpgrp, contr.treatment(6, base = 3))5 0.545115
## C(sbpgrp, contr.treatment(6, base = 3))6 0.440634
##
## Likelihood ratio test=26.58 on 5 df, p=6.879e-05
## n= 956, number of events= 723
##
## [[4]]
## Call:
## coxph(formula = Surv(survda, dead) ~ C(sbpgrp, contr.treatment(6,
##   base = 3)) + strata(sexe, agegp))
##
##               coef exp(coef) se(coef)      z
## C(sbpgrp, contr.treatment(6, base = 3))1 0.52667  1.69329  0.11807  4.461
## C(sbpgrp, contr.treatment(6, base = 3))2 0.37985  1.46206  0.10272  3.698
## C(sbpgrp, contr.treatment(6, base = 3))4 0.04290  1.04383  0.11773  0.364
## C(sbpgrp, contr.treatment(6, base = 3))5 0.09629  1.10107  0.14531  0.663
## C(sbpgrp, contr.treatment(6, base = 3))6 -0.16181  0.85060  0.28752 -0.563
##
##               p
## C(sbpgrp, contr.treatment(6, base = 3))1 8.17e-06
## C(sbpgrp, contr.treatment(6, base = 3))2 0.000218
## C(sbpgrp, contr.treatment(6, base = 3))4 0.715583
## C(sbpgrp, contr.treatment(6, base = 3))5 0.507558
## C(sbpgrp, contr.treatment(6, base = 3))6 0.573578
##
## Likelihood ratio test=30.29 on 5 df, p=1.294e-05
## n= 956, number of events= 723
##
## [[5]]
## Call:
## coxph(formula = Surv(survda, dead) ~ C(sbpgrp, contr.treatment(6,
##   base = 3)) + strata(sexe, agegp))
##
##               coef exp(coef) se(coef)      z
```



```
## C(sbpgrp, contr.treatment(6, base = 3))1 0.55054 1.73419 0.11582 4.753
## C(sbpgrp, contr.treatment(6, base = 3))2 0.36464 1.43999 0.10339 3.527
## C(sbpgrp, contr.treatment(6, base = 3))4 0.06843 1.07083 0.12012 0.570
## C(sbpgrp, contr.treatment(6, base = 3))5 0.09692 1.10177 0.14854 0.652
## C(sbpgrp, contr.treatment(6, base = 3))6 -0.15071 0.86010 0.28786 -0.524
##
## p
## C(sbpgrp, contr.treatment(6, base = 3))1 2e-06
## C(sbpgrp, contr.treatment(6, base = 3))2 0.000421
## C(sbpgrp, contr.treatment(6, base = 3))4 0.568872
## C(sbpgrp, contr.treatment(6, base = 3))5 0.514115
## C(sbpgrp, contr.treatment(6, base = 3))6 0.600589
##
## Likelihood ratio test=31.22 on 5 df, p=8.486e-06
## n= 956, number of events= 723
```

12. Pool the results for each of the five scenarios.

```
r1 <- as.vector(t(exp(summary(pool(fit1))[, c(1)])))
r2 <- as.vector(t(exp(summary(pool(fit2))[, c(1)])))
r3 <- as.vector(t(exp(summary(pool(fit3))[, c(1)])))
r4 <- as.vector(t(exp(summary(pool(fit4))[, c(1)])))
r5 <- as.vector(t(exp(summary(pool(fit5))[, c(1)])))
```

```
summary(pool(fit1))
```

```
## estimate std.error statistic
## C(sbpgrp, contr.treatment(6, base = 3))1 0.56130514 0.1239289 4.5292516
## C(sbpgrp, contr.treatment(6, base = 3))2 0.36831156 0.1061004 3.4713499
## C(sbpgrp, contr.treatment(6, base = 3))4 0.08882521 0.1246379 0.7126659
## C(sbpgrp, contr.treatment(6, base = 3))5 0.08784716 0.1571882 0.5588660
## C(sbpgrp, contr.treatment(6, base = 3))6 -0.09953872 0.2751397 -0.3617752
## df p.value
## C(sbpgrp, contr.treatment(6, base = 3))1 2336.33678 6.216963e-06
## C(sbpgrp, contr.treatment(6, base = 3))2 909.51401 5.422176e-04
## C(sbpgrp, contr.treatment(6, base = 3))4 222.32292 4.767998e-01
## C(sbpgrp, contr.treatment(6, base = 3))5 99.17335 5.775130e-01
## C(sbpgrp, contr.treatment(6, base = 3))6 2620.13541 7.175492e-01
```

This code grabs the information from the tabulated pooled results that are produced by summary. In order to make sense about these numbers, and to see what exactly is extracted in the above code, laying out the numbers in a proper table may be useful.

```
pars <- round(t(matrix(c(r1,r2,r3,r4,r5), nrow = 5)),2)
pars <- pars[, c(1, 2, 5)]
dimnames(pars) <- list(delta, c("<125", "125-140", ">200"))
pars
```

```
## <125 125-140 >200
## 0 1.75 1.45 0.91
## -5 1.69 1.41 0.90
## -10 1.70 1.43 0.86
## -15 1.77 1.50 0.87
## -20 1.74 1.43 0.86
```

All in all, it seems that even big changes to the imputations (e.g. deducting 20 mmHg) has little influence on the results. This suggests that the results are stable relatively to this type of MNAR-mechanism.

13. Perform sensitivity analysis on the mammalsleep dataset by adding and subtracting some amount from the imputed values for sws. Use `delta <- c(8, 6, 4, 2, 0, -2, -4, -6, -8)` and investigating the influence on the following regression model:

```
lm(sws ~ log10(bw) + odi)
```

Sensitivity analysis is an important tool for investigating the plausibility of the MAR assumption. We again use the δ -adjustment technique described in Van Buuren (2012, p. 185) as an informal, simple and direct method to create imputations under nonignorable models. We do so by simply adding and subtracting some amount from the imputations.

```
delta <- c(8, 6, 4, 2, 0, -2, -4, -6, -8)
ini <- mice(mammalsleep[, -1], maxit=0, print=F)
meth<- ini$meth
meth["ts"]<- "~ l(sws + ps)"
pred <- ini$pred
pred[c("sws", "ps"), "ts"] <- 0
post <- ini$post
imp.all.undamped <- vector("list", length(delta))
for (i in 1:length(delta)) {
  d <- delta[i]
  cmd <- paste("imp[[i]][, i] <- imp[[i]][, i] +", d)
  post["sws"] <- cmd
  imp <- mice(mammalsleep[, -1], meth=meth, pred=pred, post = post, maxit = 10, seed = i * 22, print=FALSE)
  imp.all.undamped[[i]] <- imp
}
output <- sapply(imp.all.undamped, function(x) pool(with(x, lm(sws ~ log10(bw) + odi)))$qbar)
cbind(delta, as.data.frame(t(output)))
```

```
## delta V1 V2 V3 V4 V5 V6 V7 V8 V9
## 1 8 NULL NULL NULL NULL NULL NULL NULL NULL
## 2 6 NULL NULL NULL NULL NULL NULL NULL NULL
## 3 4 NULL NULL NULL NULL NULL NULL NULL NULL
## 4 2 NULL NULL NULL NULL NULL NULL NULL NULL
## 5 0 NULL NULL NULL NULL NULL NULL NULL NULL
## 6 -2 NULL NULL NULL NULL NULL NULL NULL NULL
## 7 -4 NULL NULL NULL NULL NULL NULL NULL NULL
## 8 -6 NULL NULL NULL NULL NULL NULL NULL NULL
## 9 -8 NULL NULL NULL NULL NULL NULL NULL NULL
```

The estimates for different δ are not close. A clear trend for the estimates for the intercept and for bw emerges. Thus, the results are not essentially the same under all specified mechanisms and the outcomes can be deemed sensitive to the assumed mechanism.

However, in this scenario, the δ adjustment is completely unrealistic. If we look at the descriptive information for observed sws

```
summary(mammalsleep$sws)
```

```
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
## 2.100 6.250 8.350 8.673 11.000 17.900 14
```

we find that even our smallest adjustment ($\delta = |2|$) already makes up almost a quarter of the average sws. Choosing unreasonably large values may always influence your estimates. Therefore; choosing values that are reasonable given your suspicions of an assumed breach of the MAR assumption is vital.

We only used a shift parameter here. In other applications, scale or shape parameters could be more natural (see e.g. Van Buuren (2012), Ch. 3.9.1). The calculations are easily adapted to such cases.

Conclusion

We have seen that we can create multiple imputations in multivariate missing data problems that imitate deviations from MAR. The analysis used the post argument of the mice() function as a hook to alter the imputations just after they have been created by a univariate imputation function. The diagnostics shows that the trick works. The relative mortality estimates are however robust to this type of alteration.

References

Van Buuren, S. (2012), Flexible Imputation of Missing Data. Chapman & Hall/CRC, Boca Raton, FL. ISBN 9781439868249. CRC Press (<https://www.crcpress.com/product/isbn/9781439868249>), Amazon (http://www.amazon.com/Flexible-Imputation-Missing-Interdisciplinary-Statistics/dp/1439868247/ref=sr_1_1?ie=UTF8&qid=1328972069&sr=8-1).

- End of Vignette