IDA Assignment 3

Johnny Lee, s1687781

8th April 2022

Q1.

Consider the `nhanes` dataset in mice. For more information please type help(`nhanes`) in the R console.

a)

(2 marks) What percentage of the cases is incomplete?

Answer :

```
cat("The percentage of incomplete case is",
    (nrow(nhanes)-nrow(cc(nhanes)))*100/nrow(nhanes))
```

```
## The percentage of incomplete case is 48
```

b)

(4 marks) Impute the data with `mice` using the defaults with `seed=1`, in step 2 predict `bmi` from `age`, `hyp`, and `chl` by the normal linear regression model, and then pool the results. What are the proportions of variance due to the missing data for each parameter? Which parameters appear to be most affected by the nonresponse?

Answer :

```
pool1 <- pool(with(mice(nhanes, printFlag = F, seed = 1), lm(bmi ~ age + hyp + chl)))
kable(pool1$pooled[,c(1,3,7,8,10)], caption="Imputation with seed=1") %>%
  kable_styling(latex_options = "hold_position")
```

Table 1: Imputation with seed=1

term	estimate	dfcom	df	lambda
(Intercept)	19.6178925	21	16.936189	0.0893899
age	-3.5528716	21	3.528053	0.6864064
hyp	2.1970175	21	9.035495	0.3504345
chl	0.0537808	21	10.228828	0.3040806

c)

(4 marks) Repeat the analysis for seed $\in \{2, 3, 4, 5, 6\}$. Do the conclusions remain the same?

Answer :

```
pool2 <- pool(with(mice(nhanes, printFlag = F, seed = 2), lm(bmi ~ age + hyp + chl)))
pool3 <- pool(with(mice(nhanes, printFlag = F, seed = 3), lm(bmi ~ age + hyp + chl)))
pool4 <- pool(with(mice(nhanes, printFlag = F, seed = 4), lm(bmi ~ age + hyp + chl)))
pool5 <- pool(with(mice(nhanes, printFlag = F, seed = 5), lm(bmi ~ age + hyp + chl)))
pool6 <- pool(with(mice(nhanes, printFlag = F, seed = 6), lm(bmi ~ age + hyp + chl)))

parameters <- c("(Intercept)", "age", "hyp", "chl")
df <- data.frame(parameters, pool2$pooled[,10], pool3$pooled[,10],
                 pool4$pooled[,10], pool5$pooled[,10], pool6$pooled[,10])
colnames(df) <- c("parameters", "seed=2", "seed=3", "seed=4", "seed=5", "seed=6")
kable(df, caption="Imputation with seed=2,3,4,5,6") %>%
  kable_styling(latex_options = "hold_position")
```

Table 2: Imputation with seed=2,3,4,5,6

parameters	seed=2	seed=3	seed=4	seed=5	seed=6
(Intercept)	0.4144454	0.2772900	0.1315114	0.4855733	0.4168136
age	0.4033924	0.5895051	0.2189333	0.4511896	0.6549523
hyp	0.1430995	0.4101152	0.1961083	0.5942866	0.2960364
chl	0.2959966	0.5621346	0.3305334	0.2346065	0.5196295

d)

(4 marks) Repeat the analysis with $M = 100$ with the same seeds. Would you prefer these analyses over those with $M = 5$? Explain why.

Answer :

```
pool1 <- pool(with(mice(nhanes, m = 100, printFlag = F, seed = 1),
                   lm(bmi ~ age + hyp + chl)))
pool2 <- pool(with(mice(nhanes, m = 100, printFlag = F, seed = 2),
                   lm(bmi ~ age + hyp + chl)))
pool3 <- pool(with(mice(nhanes, m = 100, printFlag = F, seed = 3),
                   lm(bmi ~ age + hyp + chl)))
pool4 <- pool(with(mice(nhanes, m = 100, printFlag = F, seed = 4),
                   lm(bmi ~ age + hyp + chl)))
pool5 <- pool(with(mice(nhanes, m = 100, printFlag = F, seed = 5),
                   lm(bmi ~ age + hyp + chl)))
pool6 <- pool(with(mice(nhanes, m = 100, printFlag = F, seed = 6),
                   lm(bmi ~ age + hyp + chl)))

parameters <- c("(Intercept)", "age", "hyp", "chl")
df <- data.frame(parameters, pool1$pooled[,10], pool2$pooled[,10],
                 pool3$pooled[,10], pool4$pooled[,10],
                 pool5$pooled[,10], pool6$pooled[,10])
colnames(df) <- c("parameters", "seed=1", "seed=2", "seed=3", "seed=4",
                  "seed=5", "seed=6")
kable(df, caption="Imputation with seed=1,2,3,4,5,6 and M=100") %>%
  kable_styling(latex_options = "hold_position")
```

Table 3: Imputation with seed=1,2,3,4,5,6 and M=100

parameters	seed=1	seed=2	seed=3	seed=4	seed=5	seed=6
(Intercept)	0.2290445	0.1882474	0.2199607	0.2144722	0.2294356	0.2472607
age	0.4324680	0.4031077	0.3093072	0.3943223	0.3322570	0.4430300
hyp	0.2915346	0.2825108	0.2425105	0.2565132	0.2893046	0.2860700
chl	0.3217837	0.2939693	0.3281911	0.2835232	0.2461956	0.3113085

The pooled estimates is more stable when the value of M is higher. Thus, we would prefer $M = 100$ than $M = 5$

Q2.

(15 marks) Each of the 100 datasets contained in the file dataex2.Rdata was generated in the following way

$$y_i|x_i \overset{\text{ind.}}{\sim} \mathrm{N}(\beta_0 + \beta_1 x_i, 1), \quad x_i \overset{\text{ind.}}{\sim} \mathrm{Unif}(-1, 1), \quad \beta_0 = 1, \quad \beta_1 = 3$$

for $i = 1, \dots, 100$. Additionally, some of the responses were set to be missing using a MAR mechanism. The goal of this exercise is to study the effect that acknowledging/not acknowledging parameter uncertainty when performing step 1 of multiple imputation might have on the coverage of the corresponding confidence intervals. Further suppose that the analysis of interest in step 2 is to fit the regression model that was used to generate the data, i.e., a normal linear regression model where the response is y and the covariate is x. With the aid of the `mice` package, calculate the empirical coverage probability of the 95% confidence intervals for β_1 under the following two approaches: stochastic regression imputation and the corresponding bootstrap based version. Comment. For both approaches, please consider $m = 20$ and `seed=1`. **NOTE 1:** In order to calculate the empirical coverage probability, you only need to compute the proportion of the time (over the 100 intervals) that the interval contains the true value of the parameter. **NOTE 2:** The data are stored in an array structure such that, for instance, `dataex2[, , 1]`, corresponds to the first dataset (which has 100 rows and 2 columns, with the first column containing the values of x and the second the values of y).

Answer :

```r
# initialize a counter
count <- 0
for (i in 1:nrow(dataex2)) {
  #impute values for the ith dataset using M=20
  impute.sri <- mice(dataex2[, , i], m = 20, method = "norm.nob", printFlag = F, seed = 1)
  fit.sri <- with(impute.sri, lm(Y ~ X)) #step 2
  pool.sri <- pool(fit.sri) # step 3
  summary.sri <- summary(pool.sri, conf.int = TRUE)
  if (summary.sri[2, 7] <= 3 & summary.sri[2, 8] >= 3) {
  count <- count + 1 #add to the counter if the the value of beta1 is contained in the
  #confidence interval
  }
}
ecp.sri <- count/nrow(dataex2)
cat("the proportion of the time for Stochastic Imputation is", ecp.sri)
```

```
## the proportion of the time for Stochastic Imputation is 0.88
```

```r
# initialize a counter
count <- 0
for (i in 1:nrow(dataex2)) {
  #impute values for the ith dataset, using m=20
  impute.bootstrap <- mice(dataex2[,,i], m = 20, method = "norm.boot",
                         printFlag = FALSE, seed = 1)
  fit.bootstrap <- with(impute.bootstrap, lm(Y ~ X)) #step 2
  pool.bootstrap <- pool(fit.bootstrap) # step 3
  summary.bootstrap <- summary(pool.bootstrap, conf.int = TRUE)
  if (summary.bootstrap[2, c(7)] <= 3 & summary.bootstrap[2, c(8)] >= 3) {
    count = count + 1 #add to the counter if the true value of beta1 is contained in the
  }
}
ecp.bootstrap <- count/nrow(dataex2)
cat("the proportion of the time for Bootstrap is", ecp.bootstrap)
```

```
## the proportion of the time for Bootstrap is 0.95
```

Q3.

(9 marks) Show that for a linear (in the coefficients) regression model, the following two strategies coincide:

(i) Computing the predicted values (point estimates) from each fitted model in step 2 and then pooling them according to Rubin's rule for point estimates (i.e., averaging the predicted values across the imputed datasets).

(ii) Pooling the regression coefficients from each fitted model in step 2 using Rubin's rule for point estimates and then computing the predicted values afterwards

Answer :

We consider a linear regression model given a dataset as $\{y_i, x_{1i}, \ldots, x_{ni}\}$

$$y_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_n x_{ni} + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2)$$

Now we look into Case (i), we compute the predicted values for each fitted model from step 2. Then we obtain as below,

$$\hat{y}_i^{(m)} = \hat{\beta}_0^{(m)} + \hat{\beta}_1^{(m)} x_{1i} + \cdots + \hat{\beta}_n^{(m)} x_{ni}$$

Then we pool them according to Rubin's rule for point estimates.

$$\begin{aligned}
\bar{y}_i &= \frac{1}{M} \sum_{i=1}^{M} \hat{y}^{(m)} \\
&= \frac{1}{M} \sum_{i=1}^{M} \left(\hat{\beta}_0^{(m)} + \hat{\beta}_1^{(m)} x_{1i} + \cdots + \hat{\beta}_n^{(m)} x_{ni} \right) \\
&= \frac{1}{M} \sum_{i=1}^{M} \hat{\beta}_0^{(m)} + \frac{1}{M} \sum_{i=1}^{M} \hat{\beta}_1^{(m)} x_{1i} + \cdots + \frac{1}{M} \sum_{i=1}^{M} \hat{\beta}_n^{(m)} x_{ni} \\
&= \bar{\beta}_0 + \bar{\beta}_1 x_{1i} + \cdots + \bar{\beta}_n x_{ni}
\end{aligned}$$

Now, let us consider Case (ii) to validate if they coincide. We pool the regression coefficients from each fitted model in step 2 using Rubin's rule for point estimates.

$$\begin{aligned}
\bar{\beta}_0 &= \frac{1}{M} \sum_{i=1}^{M} \hat{\beta}_0^{(m)} \\
&\vdots \\
\bar{\beta}_n &= \frac{1}{M} \sum_{i=1}^{M} \hat{\beta}_n^{(m)}
\end{aligned}$$

Then we can compute the predicted values as follow

$$\bar{y}_i = \bar{\beta}_0 + \bar{\beta}_1 x_{1i} + \cdots + \bar{\beta}_n x_{ni}$$

As shown above, the order of the computation of predicted values for each fitted model in step 2 and pooling according to Rubin's rule for point estimates do not matter mathematically. Therefore, we conclude here by saying that both cases coincide.

Q4.

The goal of this exercise is to study different ways of using `mice` when the analysis model of interest/substantive model involves an interaction term between incomplete variables. The model used to generate the data (available in `dataex4.Rdata`), which corresponds to our model of interest in step 2, was the following one:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i} x_{2i} + \varepsilon_i, x_{1i} \overset{\text{iid}}{\sim} \mathrm{N}(0,1), \quad x_{2i} \overset{\text{iid}}{\sim} \mathrm{N}(1.5,1), \quad \varepsilon_i \overset{\text{iid}}{\sim} \mathrm{N}(0,1)$$

for $i = 1, \dots, 1000$, $\beta_0 = 1.5$, $\beta_1 = 1$, $\beta_2 = 2$ and $\beta_3 = 1$. Additionally, missingness was imposed on $y$ and $x_1$ and so the interaction variable $x_1 x_2$ also has missing values, although the missingness in this interaction variable is induced by the missing in the covariate $x_1$. In the following, please use $M = 50$ and `seed=1`.

```
kable(head(dataex4), caption="dataex4 observation")  %>%
  kable_styling(latex_options = "hold_position")
```

Table 4: dataex4 observation

y	x1	x2
NA	NA	1.0983213
4.609476	0.0265944	1.1290673
NA	-1.5165531	1.0748538
1.718231	-1.3626533	1.9411515
NA	1.1784892	0.8272496
1.820674	-0.9341513	1.9166941

a)

(6 marks) By only imputing the $y$ and $x_1$ variables in step 1, provide the estimates of β_1, β_2, and β_3 along with 95% confidence intervals. Comment. Note that this approach where the interaction variable is left outside the imputation process and calculated afterwards in the analysis model, is known as *Impute, then transform*

Answer :

```
impute.sri <- mice(dataex4, m = 50, seed = 1, printFlag = FALSE)
fit.sri <- with(impute.sri, lm(y ~ x1 + x2 + x1*x2))
pool.sri <- pool(fit.sri)
kable(summary(pool.sri, conf.int = TRUE)[, c(1,2,3,7,8)],
      caption = "Summary Statistics of Imputation of $y$ and $x_1$") %>%
  kable_styling(latex_options = "hold_position")
```

Table 5: Summary Statistics of Imputation of $y$ and $x_1$

term	estimate	std.error	2.5 %	97.5 %
(Intercept)	1.5929831	0.0954133	1.404501	1.7814655
x1	1.4112333	0.0973291	1.219397	1.6030697
x2	1.9658191	0.0532322	1.860657	2.0709812
x1:x2	0.7550367	0.0570146	0.642302	0.8677715

b)

(10 marks) Now, start by calculating the interaction variable in the incomplete data and append it as a variable to your dataset. Then, use *passive imputation* to impute the interaction variable. Provide the estimates of β_1, β_2, and β_3 along with 95% confidence intervals. Comment.

Answer :

```
x1 <- dataex4$x1; x2 <- dataex4$x2; dataex4$x1x2 <- x1*x2
impute.null <- mice(dataex4, maxit = 0)
method <- impute.null$method
method["x1x2"] <- "~I(x1*x2)"
pred <- impute.null$predictorMatrix
pred[c("x1", "x2"), "x1x2"] <- 0
visit.seq <- impute.null$visitSequence
visit.seq
```

```
## [1] "y"    "x1"   "x2"   "x1x2"
```

```
impute.passive <- mice(dataex4, method = method, predictorMatrix = pred,
                visitSequence = visit.seq, m = 50, seed = 1, printFlag = FALSE)
pool.passive <- pool(with(impute.passive, lm(y ~ x1 + x2 + x1*x2)))
kable(summary(pool.passive, conf.int=TRUE)[,c(1,2,3,7,8)],
      caption = "Summary Statistics of Imputation of $y$ and $x_1$") %>%
  kable_styling(latex_options = "hold_position")
```

Table 6: Summary Statistics of Imputation of $y$ and $x_1$

term	estimate	std.error	2.5 %	97.5 %
(Intercept)	1.5534782	0.0884221	1.3788626	1.7280939
x1	1.1926170	0.0958435	1.0034980	1.3817360
x2	1.9964402	0.0493658	1.8989468	2.0939336
x1:x2	0.8740573	0.0567852	0.7615712	0.9865434

c)

(10 marks) Now that you have already appended the interaction variable to the dataset, impute it as it was *just another variable* (or like any other variable) in the dataset and use this variable for the interaction term in step 2. Provide the estimates of β_1, β_2 and β_3 along with 95% confidence intervals. Comment.

Answer :

```
impute.jav <- mice(dataex4, m = 50, seed = 1, printFlag = FALSE)
fit.jav <- with(impute.jav, lm(y ~ x1 + x2 + x1x2))
pool.jav <- pool(fit.jav)
kable( summary(pool.jav, conf.int=TRUE)[, c(1,2,3,7,8)],
      caption = "Summary Statistics of Imputation of $y$ and $x_1$") %>%
  kable_styling(latex_options = "hold_position")
```

Table 7: Summary Statistics of Imputation of $y$ and $x_1$

term	estimate	std.error	2.5 %	97.5 %
(Intercept)	1.499714	0.0782144	1.3452011	1.654227
x1	1.003930	0.0822837	0.8414967	1.166363
x2	2.026180	0.0437161	1.9398113	2.112548
x1x2	1.017793	0.0442807	0.9303479	1.105238

d)

(6 marks) What is the obvious conceptual drawback of the *just another variable* approach for imputing interactions?

Answer :

Q5

(30 marks) The file NHANES2.Rdata contains a subset of data from the *National Health and Nutrition Examination Survey* (NHANES), whose goal is to assess the health and nutritional status of adults and children in the United States. The variables in the dataset are the following:

- wgt: weight in kg,
- gender: male vs female,
- bili: bilirubin concentration in mg/dL,
- age: in years,
- chol: total serum cholestrol in mg/dL,
- HDL: High-density lipoprotein cholestrol in mg/dL,
- hgt: height in metres,
- educ: educational status; 5 ordered categories,
- race: 5 unordered categories,
- SBP: systolic blood pressure in mmHg,
- hypten: hyptertensive status; binary,
- WC: waist circumference in cm.

The analysis of interest is the following:

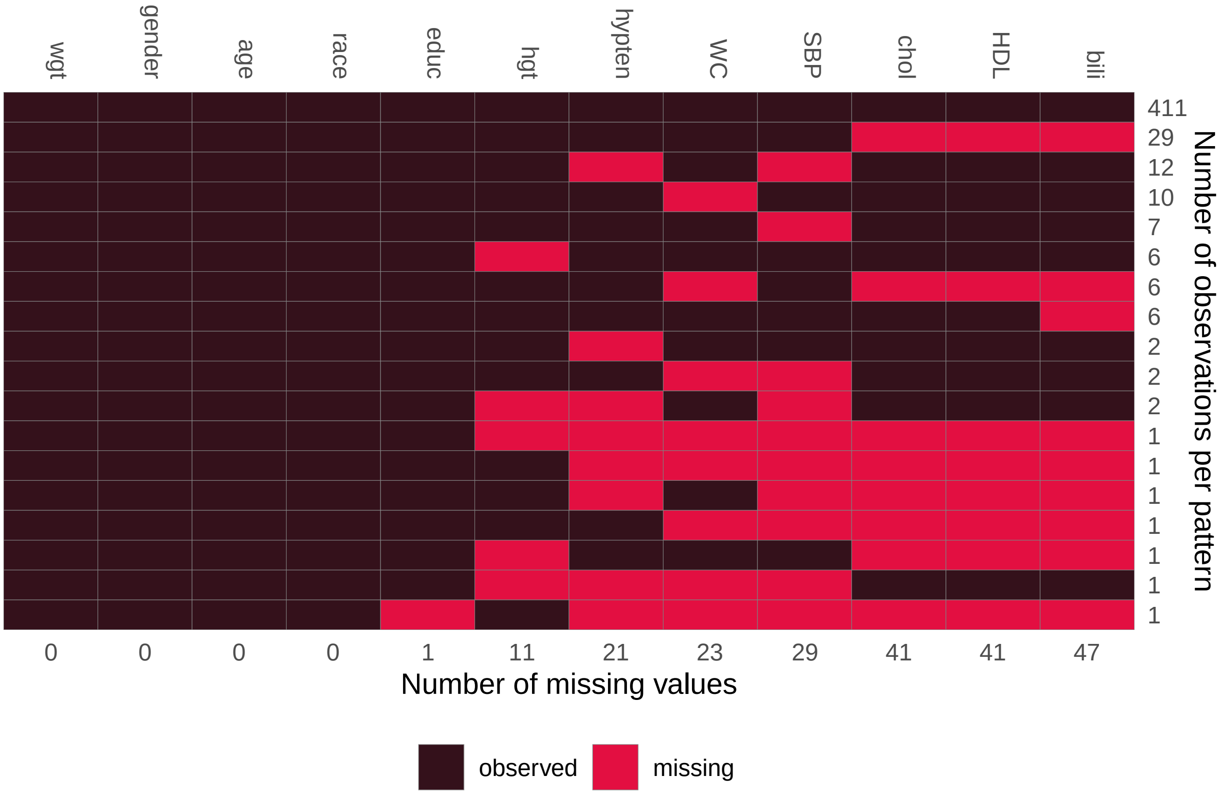
$$\text{wgt} = \beta_0 + \beta_1\text{gender} + \beta_2\text{age} + \beta_3\text{hgt} + \beta_4\text{WC} + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2).$$

Using multiple imputation and conducting all necessary checks, report your findings.

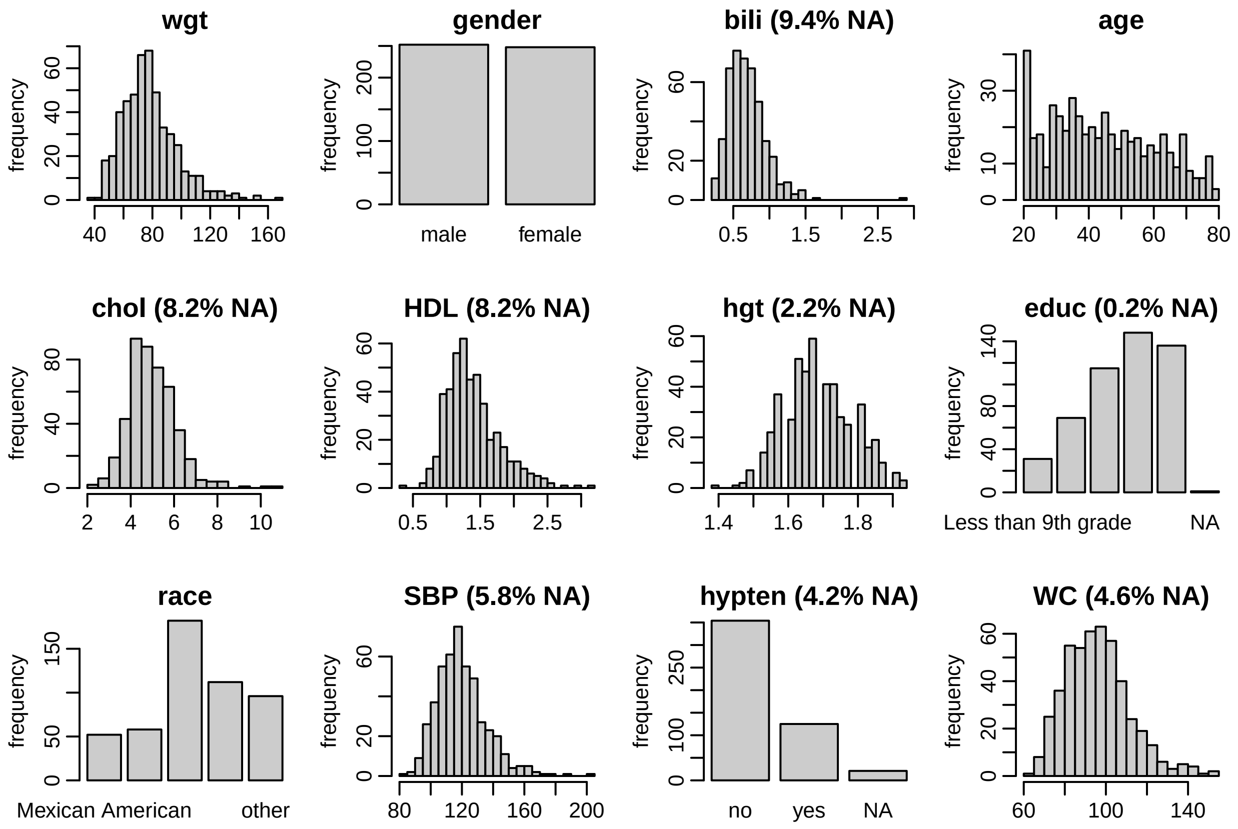
Answer :

```
nhanes2 <- NHANES2
```

```
md_pattern(nhanes2, pattern = FALSE, color = c('#34111b', '#e30f41'))
```



```
par(mar = c(3,3,2,1), mgp = c(2,0.6,0))
plot_all(nhanes2, breaks = 30, ncol = 4)
```



```
impute.null <- mice(NHANES, maxit = 0)
impute.null
```

```
## Class: mids
## Number of multiple imputations:  5
## Imputation methods:
##       SBP    gender       age      race        WC       alc      educ     creat
##        ""        ""        ""        ""     "pmm"  "logreg"        ""     "pmm"
##      albu  uricacid      bili     occup     smoke
##     "pmm"     "pmm"     "pmm" "polyreg"    "polr"
## PredictorMatrix:
##        SBP gender age race WC alc educ creat albu uricacid bili occup smoke
## SBP      0      1   1    1  1   1    1     1    1        1    1     1     1
## gender   1      0   1    1  1   1    1     1    1        1    1     1     1
## age      1      1   0    1  1   1    1     1    1        1    1     1     1
## race     1      1   1    0  1   1    1     1    1        1    1     1     1
## WC       1      1   1    1  0   1    1     1    1        1    1     1     1
## alc      1      1   1    1  1   0    1     1    1        1    1     1     1
```

We have 411 observations with observed values on all 12 variables. Also, 10 observations for which on WC is missing and 7 observations for which on hgt is missing. Interestingly, we can observe that there are 29 observations for which on chol, HDL and bili are missing.