Incomplete Data Analysis: Assignment 3

Xiao Heng (s2032451)

1.

(a)

```
n1 <- dim(nhanes)[1]
sum(is.na(rowSums(nhanes)))/n1
```

```
## [1] 0.48
```

(b)

```
imps1b <- mice(nhanes, printFlag = FALSE, seed = 1)
fits1b <- with(imps1b, lm(bmi ~ age + hyp + chl))
ests1b <- pool(fits1b)
summary(ests1b, conf.int = TRUE)[, c(2, 3, 6, 7, 8)]
```

```
##      estimate  std.error      p.value        2.5 %
## 1 19.61789252 3.41003531 2.376938e-05 12.421281424
## 2 -3.55287155 1.54113006 9.129146e-02 -8.067205920
## 3  2.19701748 2.10797844 3.243838e-01 -2.568707058
## 4  0.05378081 0.02031792 2.401044e-02  0.008646559
##        97.5 %
## 1 26.81450361
## 2  0.96146282
## 3  6.96274202
## 4  0.09891506
```

After modelling, the proportions of variance due to the missing data for each parameter are computed as below:

```
pvar1b <- t(ests1b$pooled['lambda'])[1:4]
names(pvar1b) <- c("intercept", "age", "hyp", "chl")
pvar1b
```

```
##  intercept        age        hyp        chl
## 0.08938989 0.68640637 0.35043452 0.30408063
```

From the result, we could know that the `age` term is most affected by the nonresponse.

(c)

```
pvars1c <- matrix(nrow = 5, ncol = 4)
dimnames(pvars1c) <- list(c(2:6), c("intercept", "age", "hyp", "chl"))
for(i in 2:6)
{
  imps1c <- mice(nhanes, printFlag = FALSE, seed = i)
  fits1c <- with(imps1c, lm(bmi ~ age + hyp + chl))
  ests1c <- pool(fits1c)
  print(summary(ests1c, conf.int = TRUE)[, c(2, 3, 6, 7, 8)])
  pvars1c[i-1, ] <- t(ests1c$pooled['lambda'])
}
```

```
##     estimate  std.error     p.value       2.5 %     97.5 %
## 1 19.9464142 4.36349993 0.002083932  9.79011819 30.1027102
## 2 -4.0615093 1.27092690 0.013061972 -7.00354844 -1.1194702
## 3  1.5304762 2.01529855 0.459204412 -2.75985082  5.8208033
## 4  0.0628349 0.02215369 0.016956575  0.01376007  0.1119097
##      estimate  std.error     p.value       2.5 %     97.5 %
## 1 20.55844343 3.97226884 0.000307981 11.81327504 29.3036118
## 2 -3.85753338 1.52845844 0.056159642 -7.86725134  0.1521846
## 3  1.35281238 2.19759517 0.555952054 -3.75116261  6.4567874
## 4  0.05872834 0.02727828 0.083285020 -0.01114438  0.1286011
##      estimate  std.error      p.value       2.5 %
## 1 19.39540373 3.57665642 6.168639e-05 11.79704747
## 2 -3.50603350 1.06606899 6.017910e-03 -5.81391990
## 3  2.75053046 2.00975193 1.935191e-01 -1.57590349
## 4  0.04920611 0.02049964 3.842988e-02  0.00322429
##        97.5 %
## 1 26.99375999
## 2 -1.19814710
## 3  7.07696442
## 4  0.09518792
##      estimate  std.error     p.value       2.5 %     97.5 %
## 1 19.17135935 4.72035771 0.006088277  7.73332762 30.6093911
## 2 -3.49672250 1.34518423 0.036037680 -6.68976726 -0.3036777
## 3  1.50954775 2.95726023 0.633158537 -6.28293447  9.3020300
## 4  0.06081272 0.02086271 0.012719752  0.01545955  0.1061659
##      estimate  std.error     p.value       2.5 %     97.5 %
## 1 20.52083805 4.20049902 0.001438187 10.73223644 30.3094397
## 2 -2.92141353 1.47485121 0.120891488 -7.06709776  1.2242707
## 3  1.22474596 2.12883012 0.577257293 -3.49110047  5.9405924
## 4  0.04949218 0.02534323 0.101257260 -0.01333935  0.1123237
```

```
pvars1c
```

```
##   intercept       age       hyp       chl
## 2 0.4144454 0.4033924 0.1430995 0.2959966
## 3 0.2772900 0.5895051 0.4101152 0.5621346
## 4 0.1315114 0.2189333 0.1961083 0.3305334
## 5 0.4855733 0.4511896 0.5942866 0.2346065
## 6 0.4168136 0.6549523 0.2960364 0.5196295
```

With different seeds, the result is not stable anymore, since in some cases, the `age` term is not the one with largest proportion of variance due to missingness anymore.

Note that, since here we adopt the default `m=5`, it may not converge. But we don't apply the convergence because in the following question we will increase the value of `m`, and we will also do this in the final section. In this case, the unstable result here almost indicates the unconvergence.

(d)

```r
pvars1d <- matrix(nrow = 5, ncol = 4)
dimnames(pvars1d) <- list(c(2:6), c("intercept", "age", "hyp", "chl"))
for(i in 2:6)
{
  imps1d <- mice(nhanes, m = 100, printFlag = FALSE, seed = i)
  fits1d <- with(imps1d, lm(bmi ~ age + hyp + chl))
  ests1d <- pool(fits1d)
  print(summary(ests1d, conf.int = TRUE)[, c(2, 3, 6, 7, 8)])
  pvars1d[i-1, ] <- t(ests1d$pooled['lambda'])
}
```

```
##     estimate  std.error      p.value        2.5 %
## 1 20.4677977 3.63913732 4.241712e-05 12.734541608
## 2 -3.6337887 1.26258958 1.468267e-02 -6.404409963
## 3  1.7199145 2.14123878 4.355947e-01 -2.883346989
## 4  0.0535212 0.02123996 2.511358e-02  0.007784548
##        97.5 %
## 1 28.20105375
## 2 -0.86316750
## 3  6.32317603
## 4  0.09925785
##      estimate  std.error      p.value        2.5 %
## 1 20.37418806 3.80183492 8.133609e-05 12.266333190
## 2 -3.55706093 1.19140872 1.043059e-02 -6.128410634
## 3  1.55756211 2.10033700 4.702202e-01 -2.933905413
## 4  0.05445409 0.02212929 2.894663e-02  0.006552328
##       97.5 %
## 1 28.4820429
## 2 -0.9857112
## 3  6.0490296
## 4  0.1023558
##      estimate  std.error      p.value        2.5 %
## 1 20.38340913 3.74422202 6.757562e-05 12.403525217
## 2 -3.64815327 1.24485339 1.315984e-02 -6.374986681
## 3  1.65946951 2.11584235 4.457676e-01 -2.873243617
## 4  0.05511595 0.02121066 2.139076e-02  0.009510518
##       97.5 %
## 1 28.3632930
## 2 -0.9213199
## 3  6.1921826
## 4  0.1007214
##      estimate  std.error      p.value       2.5 %
## 1 20.29458168 3.74163945 7.527037e-05 12.30610203
## 2 -3.76297245 1.19498949 7.902411e-03 -6.35139064
## 3  1.80283168 2.10369937 4.063846e-01 -2.72408399
## 4  0.05534382 0.02056097 1.721670e-02  0.01135482
##        97.5 %
## 1 28.28306133
## 2 -1.17455426
## 3  6.32974735
## 4  0.09933281
##      estimate  std.error     p.value        2.5 %
## 1 20.26848393 3.82810287 0.000104003 12.077377732
## 2 -3.59309185 1.30454950 0.019496731 -6.481208721
## 3  1.86219629 2.13127433 0.397434338 -2.721944388
## 4  0.05319323 0.02159122 0.028364041  0.006579902
##        97.5 %
## 1 28.45959013
## 2 -0.70497499
## 3  6.44633696
## 4  0.09980655
```

```
pvars1d
```

```
##   intercept       age       hyp       chl
## 2 0.1882474 0.4031077 0.2825108 0.2939693
## 3 0.2199607 0.3093072 0.2425105 0.3281911
## 4 0.2144722 0.3943223 0.2565132 0.2835232
## 5 0.2294356 0.3322570 0.2893046 0.2461956
## 6 0.2472607 0.4430300 0.2860700 0.3113085
```

2.

```
n2 <- length(dataex2[1, 1, ])
param2nob <- param2boot <- rep(NA, n2)
i=1
for(i in 1:n2)
{
  # not acknowledged parameter uncertainty
  imps2nob <- mice(dataex2[, , i], m = 20, method = "norm.nob", printFlag = FALSE, seed = 1)
  fits2nob <- with(imps2nob, lm(Y ~ X))
  ests2nob <- pool(fits2nob)
  param2nob[i] <- ests2nob$pooled[2, 3]
  # acknowledged parameter uncertainty
  imps2boot <- mice(dataex2[, , i], m = 20, method = "norm.boot", printFlag = FALSE, seed = 1)
  fits2boot <- with(imps2boot, lm(Y ~ X))
  ests2boot <- pool(fits2boot)
  param2boot[i] <- ests2boot$pooled[2, 3]
}
quantile(param2nob, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 2.556846 3.444609
```

```
quantile(param2boot, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 2.552051 3.422950
```

With more copies (`m = 100`), under different seeds, the results turn to be more stable and consistent, indicating that the `age` term is indeed the one with largest proportion of variance due to missingness, and so that been most affected.

3.

For the first strategy, under the 1-covariate condition (only β_0 for constant and β_1 for single covariate), the predicted values from each fitted model are given by:

$$\hat{y}_{mis}^{(m)} = \hat{\beta_0}^{(m)} + X_{mis}\hat{\beta_1}^{(m)} + z^{(m)},\ z^{(m)} \sim N(0, (\hat{\sigma}^{(m)})^2)$$

Then following the Robin's rule, the pooled point estimates are:

$$\hat{y}_{mis} = \frac{1}{M}\sum_{m=1}^{M}\hat{y}_{mis}^{(m)}$$

While for the second strategy, we pool the regression coefficients ($\hat{\theta} = \{\hat{\beta}, \hat{\sigma}\}$) from each fitted model as:

$$\hat{\theta} = \frac{1}{M}\sum_{m=1}^{M}\hat{\theta}^{(m)}$$

Then compute the predicted values:

$$\hat{y}_{mis} = \hat{\beta}_0 + X_{mis}\hat{\beta}_1 + z,\ \ z \sim N(0, \hat{\sigma}^2)$$

Actually, these two strategies are coincide, proved by:

$$\begin{aligned}
\hat{y}_{mis} &= \frac{1}{M}\sum_{m=1}^{M}\hat{y}_{mis}^{(m)} \\
&= \frac{1}{M}\sum_{m=1}^{M}[\hat{\beta_0}^{(m)} + X_{mis}\hat{\beta_1}^{(m)} + z^{(m)}],\ \ z^{(m)} \sim N(0, (\hat{\sigma}^{(m)})^2) \\
&= \frac{1}{M}\sum_{m=1}^{M}\hat{\beta_0}^{(m)} + \frac{X_{mis}}{M}\sum_{m=1}^{M}\hat{\beta_1}^{(m)} + \frac{1}{M}\sum_{m=1}^{M}z^{(m)} \\
&= \hat{\beta}_0 + X_{mis}\hat{\beta}_1 + z,\ \ z \sim N(0, \hat{\sigma}^2)
\end{aligned}$$

Now, when it is extended to p-covariates condition, the process is similar, just substituting the β_1 into β_1 to β_p, or β_{-0} (note that now X_{mis} is not just vector but a matrix).

4.

(a)

```
imps4a <- mice(dataex4, printFlag = FALSE, seed = 1)
long <- mice::complete(imps4a, "long", include = TRUE)
long$x1x2 <- with(long, x1*x2)
imps4a <- as.mids(long)
visSeq <- imps4a$visitSequence
visSeq
```

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

```
fits4a <- with(imps4a, lm(y ~ x1 + x2 + x1x2))
ests4a <- pool(fits4a)
beta4a <- summary(ests4a, conf.int = TRUE)[c(2:4), c(2, 7, 8)]
rownames(beta4a) <- c("beta1", "beta2", "beta3")
beta4a
```

```
##        estimate     2.5 %    97.5 %
## beta1 1.3988010 1.0910130 1.7065889
## beta2 1.9938228 1.8909276 2.0967180
## beta3 0.7611631 0.6212709 0.9010553
```

Under the *Impute, then transform* method, by leaving any derived data outside the imputation process, now we can deal with the "interaction". With this measure, the x_1x_2 term would be consistent, and the main problem of this method is exact the unused interacting term during the imputation process, leading to the bias of parameter estimator of the x_1x_2 term towards 0. Comparing with the "true" parameters which are used during the simulation, $\beta_1 = 1, \ \beta_2 = 2$ and $\beta_3 = 1$ (in simulation, $\beta_0 = 1.5$, but this is not required to analyze under the requirement of the assignment, so we only focus on the other three parameters). From the result, the estimated $\hat{\beta}_2$ is very closed to the true value, while the estimation of $\hat{\beta}_1$ and $\hat{\beta}_3$ is not good enough ($\hat{\beta}_1$ is over-estimated while $\hat{\beta}_3$ is under-estimated), and their corresponding 95% confidence intervals could just marginally contain (or around) the true values.

(b)

```
x1x2 <- dataex4$x1 * dataex4$x2
dataex4b <- cbind(dataex4, x1x2)
m=make.method(dataex4b)
m["x1x2"] <- "~I(x1*x2)"
p=make.predictorMatrix(dataex4b)
p[c("x1", "x2"), "x1x2"] <- 0
imps4b <- mice(dataex4b, meth = m, pred = p, printFlag = FALSE, seed = 1)
fits4b <- with(imps4b, lm(y ~ x1 + x2 + x1x2))
ests4b <- pool(fits4b)
beta4b <- summary(ests4b, conf.int = TRUE)[c(2:4), c(2, 7, 8)]
rownames(beta4b) <- c("beta1", "beta2", "beta3")
beta4b
```

```
##        estimate     2.5 %    97.5 %
## beta1 1.2187504 0.9266141 1.5108868
## beta2 1.9985444 1.9139246 2.0831641
## beta3 0.8505878 0.7558569 0.9453187
```

With the *Passive imputation* method, we append the calculated interaction term in the original dataset, and the transformation is done within the imputation process. In this case, this method removes the bias from the previous algorithm above. While from the result, it is indeed much better than the previous one, with less bias and more narrow confidence intervals. However, the 95% confidence intervals of $\hat{\beta}_3$ term still does not contain the true value of 1, just moves closer.

(c)

```
imps4c <- mice(dataex4b, printFlag = FALSE, seed = 1)
fits4c <- with(imps4c, lm(y ~ x1 + x2 + x1x2))
ests4c <- pool(fits4c)
beta4c <- summary(ests4c, conf.int = TRUE)[c(2:4), c(2, 7, 8)]
rownames(beta4c) <- c("beta1", "beta2", "beta3")
beta4c
```

```
##        estimate     2.5 %   97.5 %
## beta1 0.9721571 0.7702461 1.174068
## beta2 2.0200797 1.8967237 2.143436
## beta3 1.0291156 0.9351433 1.123088
```

With the *Just another variable (JAV)* method, or under the name *Transform, then impute*, we compute the x_1x_2 before imputation, and treat it as the same as others (although this may cause additional linear dependencies). However, among the results from these three methods, the estimates by this method perform the best, with all three parameter estimators very close to the true value (which are also all contained within the 95% confidence intervals).

(d)

The obvious conceptual drawback of the Just Another Variable approach for imputing interactions is that, it treats the interacting term (in this example the x_1x_2 term) as a common variable as others, indicating the omission of its internal construction or relationship with other terms. Without the generative information, not utilizing the information from other observed values, this approach depends on congenial models such as the multivariate normal, which are usually strongly mis-specified, and this procedure is only unbiased relies on the MCAR assumption (in other words, the consistency cannot be secured).

5.

In this part, we analyze the missingness and multiple imputation on the 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 analysis of interest is the following:

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

First, we briefly inspect the dimension of the data, finding that there are 500 rows and 12 variables.

```r
dim(NHANES2)
```

```
## [1] 500  12
```

The further check the nature of variables and the coded rules. Besides normal number type `num`, we notice that there are three factors `Factor` as well as one ordered factor `Ord.factor`.

```r
str(NHANES2)
```

```
## 'data.frame':    500 obs. of  12 variables:
##  $ wgt   : num  78 78 75.3 90.7 112 ...
##  $ gender: Factor w/ 2 levels "male","female": 1 1 2 1 2 1 2 2 1 1 ...
##  $ bili  : num  1.1 0.7 0.5 0.8 0.6 0.7 1.1 0.8 0.8 0.5 ...
##  $ age   : num  67 39 64 36 33 62 56 63 55 20 ...
##  $ chol  : num  6.13 4.65 4.14 3.47 6.31 4.47 6.41 5.51 7.01 3.75 ...
##  $ HDL   : num  1.09 1.14 1.29 1.37 1.27 0.85 1.81 2.38 2.79 1.03 ...
##  $ hgt   : num  1.75 1.78 1.63 1.93 1.73 ...
##  $ educ  : Ord.factor w/ 5 levels "Less than 9th grade"<..: 5 3 5 4 4 3 4 5 4 2 ...
##  $ race  : Factor w/ 5 levels "Mexican American",..: 5 3 5 3 4 5 4 5 3 3 ...
##  $ SBP   : num  139 103 NaN 115 107 ...
##  $ hypten: Factor w/ 2 levels "no","yes": 2 1 2 2 1 2 NA 1 2 1 ...
##  $ WC    : num  91.6 84.5 91.6 95.4 119.6 ...
```

The information about "min/max/mean/quantiles/missingness" could be also easily obtained by summary command. Within our interest (the formula above), only two variables `hgt` and `WC` hace missing values.

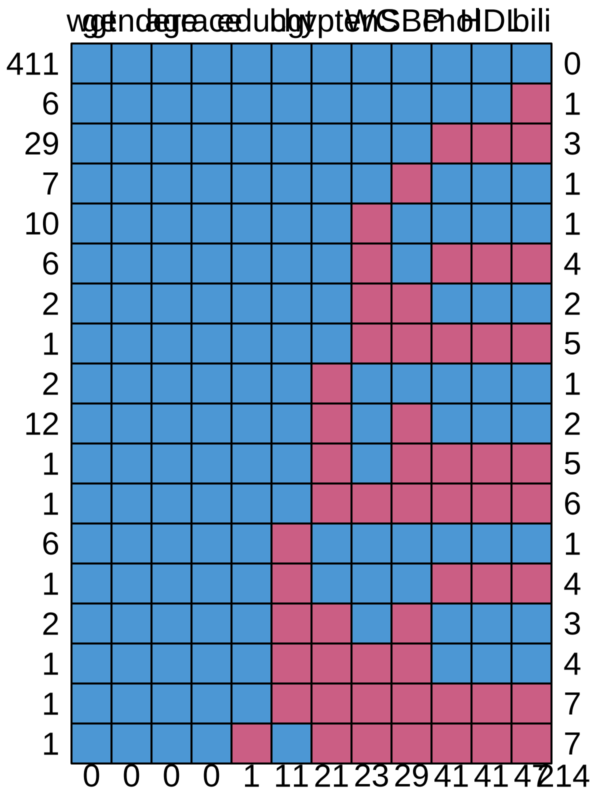
```r
summary(NHANES2)[, 1:6]; summary(NHANES2)[, 7:12]
```

```
##       wgt             gender          bili       
##  Min.   : 39.01   male  :252   Min.   :0.2000  
##  1st Qu.: 65.20   female:248   1st Qu.:0.6000  
##  Median : 76.20                Median :0.7000  
##  Mean   : 78.25                Mean   :0.7404  
##  3rd Qu.: 86.41                3rd Qu.:0.9000  
##  Max.   :167.38                Max.   :2.9000  
##                                NA's   :47      
##       age             chol            HDL
##  Min.   :20.00   Min.   : 2.07   Min.   :0.360
##  1st Qu.:31.00   1st Qu.: 4.27   1st Qu.:1.110
##  Median :43.00   Median : 4.86   Median :1.320
##  Mean   :45.02   Mean   : 5.00   Mean   :1.395
##  3rd Qu.:58.00   3rd Qu.: 5.64   3rd Qu.:1.590
##  Max.   :79.00   Max.   :10.68   Max.   :3.130
##                  NA's   :41      NA's   :41
```

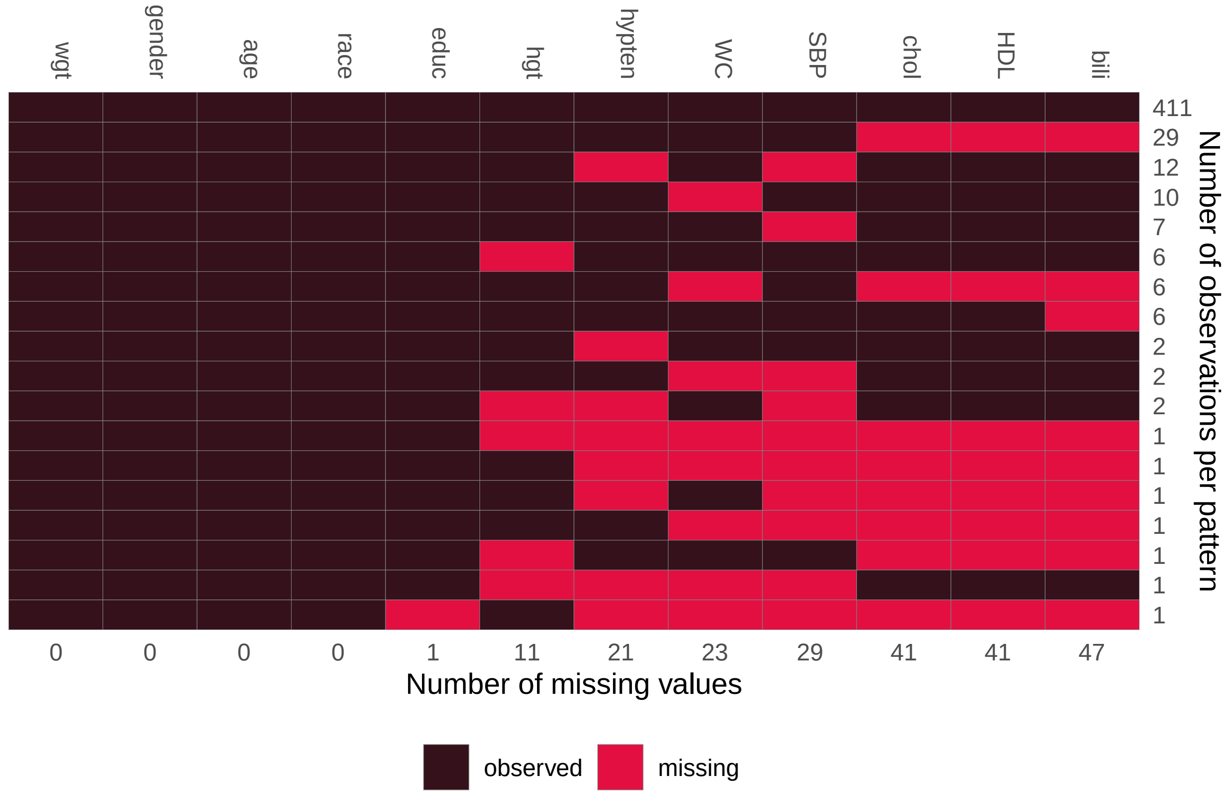
```
##       hgt                            educ
##  Min.   :1.397   Less than 9th grade : 31
##  1st Qu.:1.626   9-11th grade        : 69
##  Median :1.676   High school graduate:115
##  Mean   :1.687   some college        :148
##  3rd Qu.:1.753   College or above    :136
##  Max.   :1.930   NA's                :  1
##  NA's   :11
##                  race          SBP          hypten
##  Mexican American  : 52   Min.   : 81.33   no  :354
##  Other Hispanic    : 58   1st Qu.:109.00   yes :125
##  Non-Hispanic White:182   Median :118.67   NA's: 21
##  Non-Hispanic Black:112   Mean   :120.05
##  other             : 96   3rd Qu.:128.67
##                           Max.   :202.00
##                           NA's   :29
##        WC
##  Min.   : 61.90
##  1st Qu.: 84.80
##  Median : 95.00
##  Mean   : 96.07
##  3rd Qu.:104.80
##  Max.   :154.70
##  NA's   :23
```

Visualize the missingness pattern in two ways (actually the similar results).

```
mdpat_mice <- md.pattern(NHANES2)
```



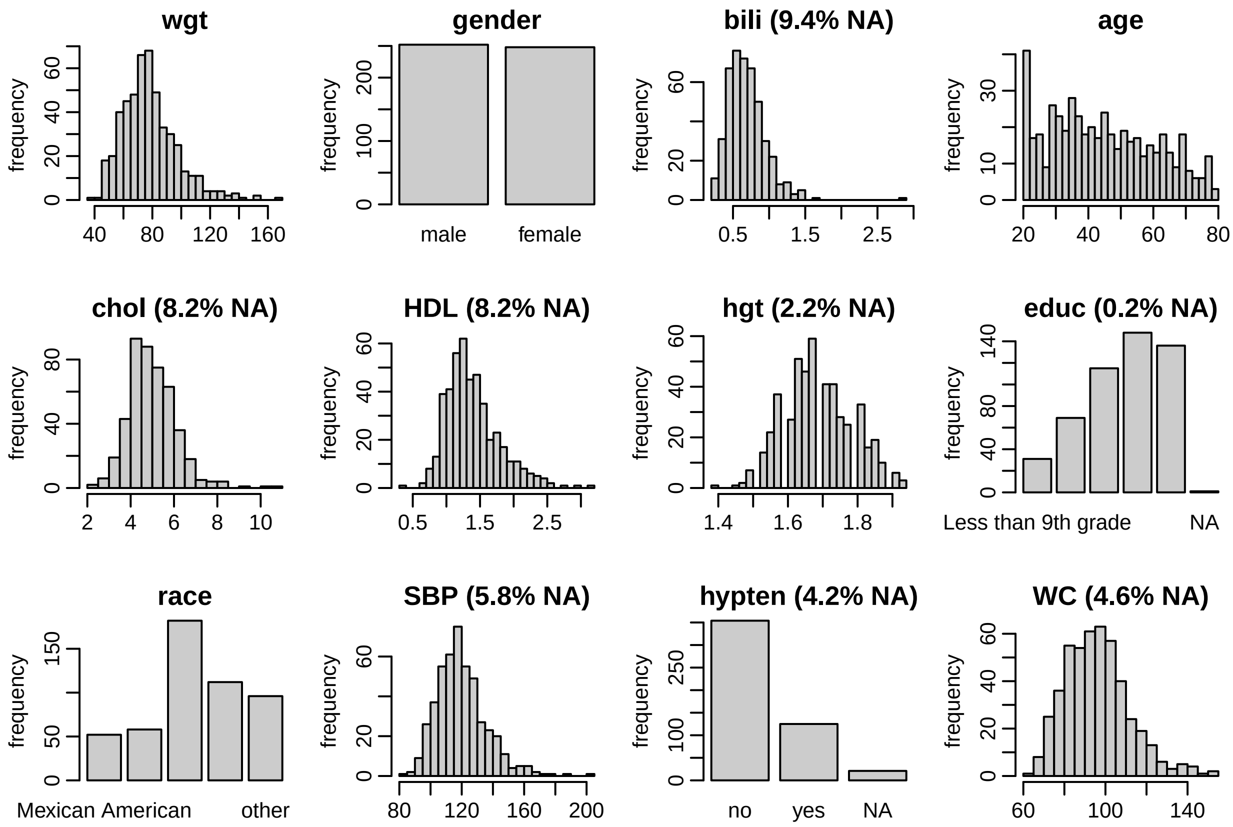
```
require(JointAI)
md_pattern(NHANES2, pattern = FALSE, color = c('#34111b', '#e30f41'))
```



Now we could conclude that there are 411 observations with completely observed values among all variables.

Since predictive mean matching `pmm` is the default method in `mice` for continuous variables, we still could have a further check based on the observed data distribution (e.g., check whether the normality assumption is roughly met) and assign a more suitable method for imputing. In most cases, the continuous variable distributions are skewed, indicating that the predictive mean matching is already the best option.

```
par(mar = c(3, 3, 2, 1), mgp = c(2, 0.6, 0))
plot_all(NHANES2, breaks = 30, ncol = 4)
```



Now we could proceed to the imputation, starting with a setup step to modify the model settings.

```
imp5 <- mice(NHANES2, maxit = 0)
imp5
```

```
## Class: mids
## Number of multiple imputations:  5
## Imputation methods:
##      wgt   gender     bili      age     chol      HDL
##       ""       ""    "pmm"       ""    "pmm"    "pmm"
##      hgt     educ     race      SBP   hypten       WC
##    "pmm"   "polr"       ""    "pmm" "logreg"    "pmm"
## PredictorMatrix:
##        wgt gender bili age chol HDL hgt educ race SBP
## wgt      0      1    1   1    1   1   1    1    1   1
## gender   1      0    1   1    1   1   1    1    1   1
## bili     1      1    0   1    1   1   1    1    1   1
## age      1      1    1   0    1   1   1    1    1   1
## chol     1      1    1   1    0   1   1    1    1   1
## HDL      1      1    1   1    1   0   1    1    1   1
##        hypten WC
## wgt         1  1
## gender      1  1
## bili        1  1
## age         1  1
## chol        1  1
## HDL         1  1
```

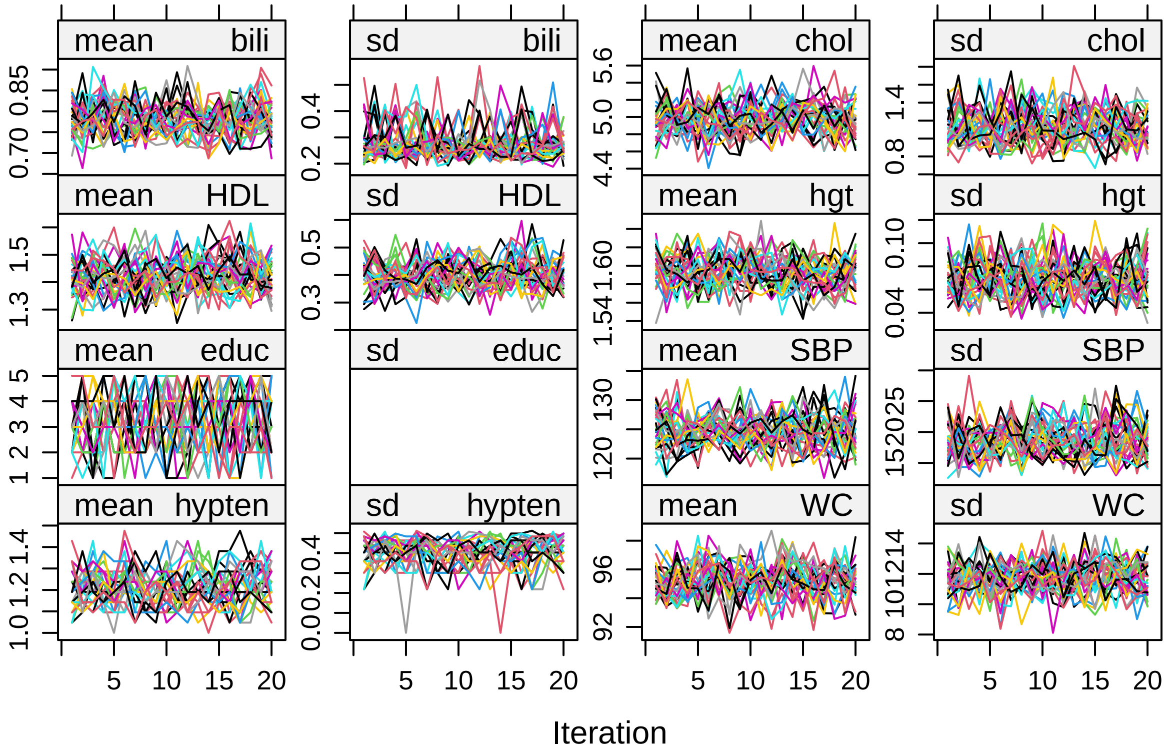
Since there is no derived or interacted variable, there is no need to modify `predictorMatrix`. Also, due to the skewness, there is no need to set certain term with the `norm` in `method` (**note that the `hgt` term seems really close to normal distribution, but with specific missing values at some locations, leading to a bad continuous approximation. Since in the lecture's example, this term is also not regarded as a "norm", here we choose to maintain its method as "pmm".**). Now, for the formal imputation step, we set `maxit=20` and `M=30`.

```
imps5 <- mice(NHANES2, maxit = 20, m = 30, seed = 1, printFlag = FALSE)
imps5$loggedEvents
```

```
## NULL
```

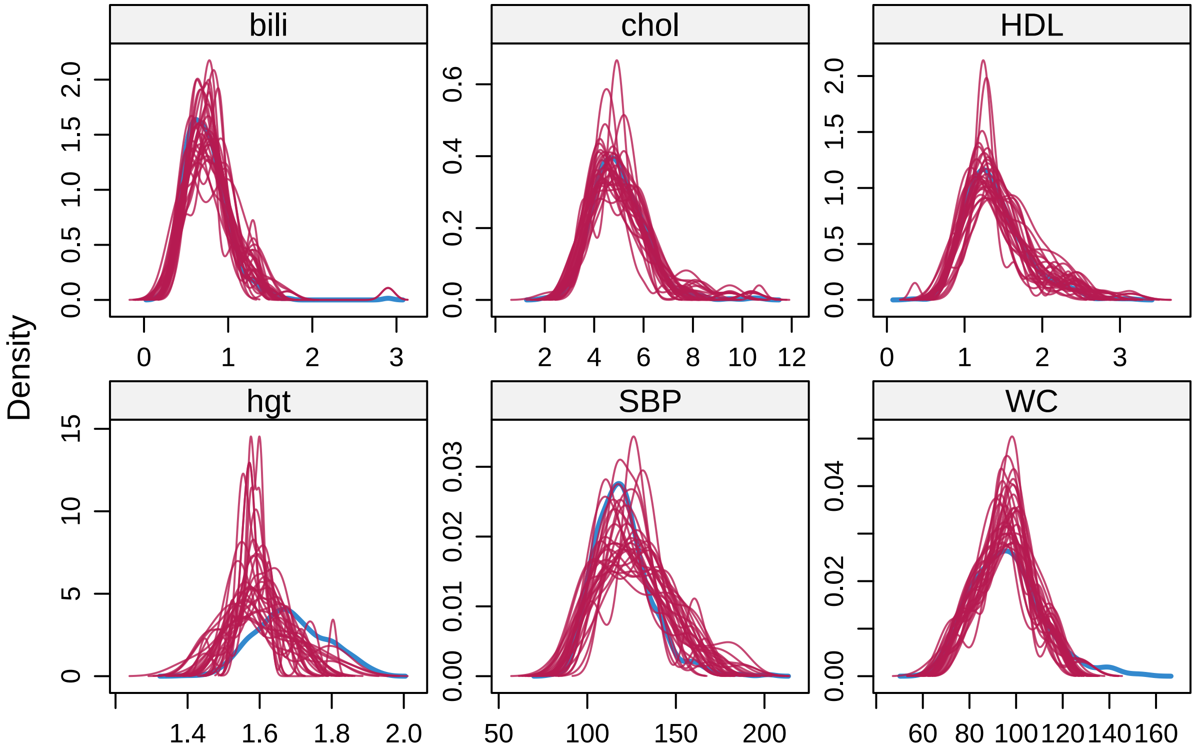
To check the convergence, we need to look at the chains of imputed values.

```
plot(imps5, layout = c(4,4))
```



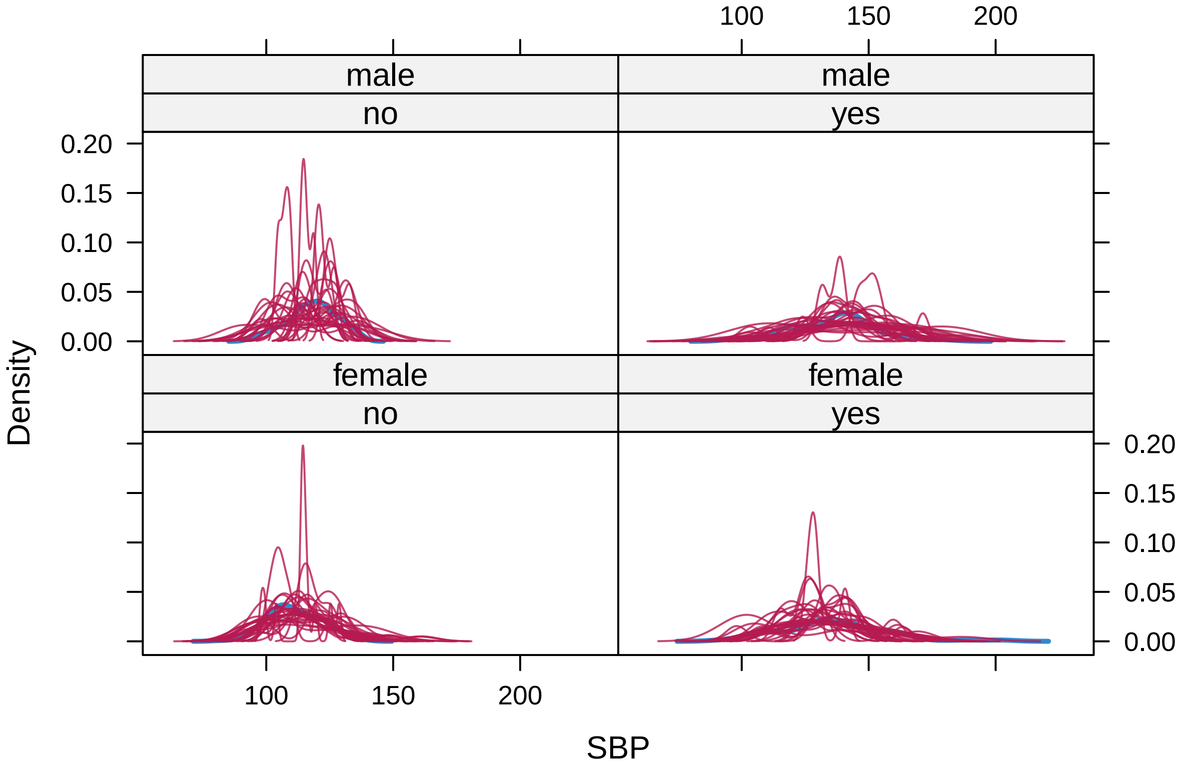
From the graph, all variables seem good regarding the convergence. Note that, there is a blank subplot, associating with the standard deviation of `educ`. This is due to the fact that there is only 1 missing value in this variable. Now we check the fitting of imputed distributions.

```
densityplot(imps5)
```

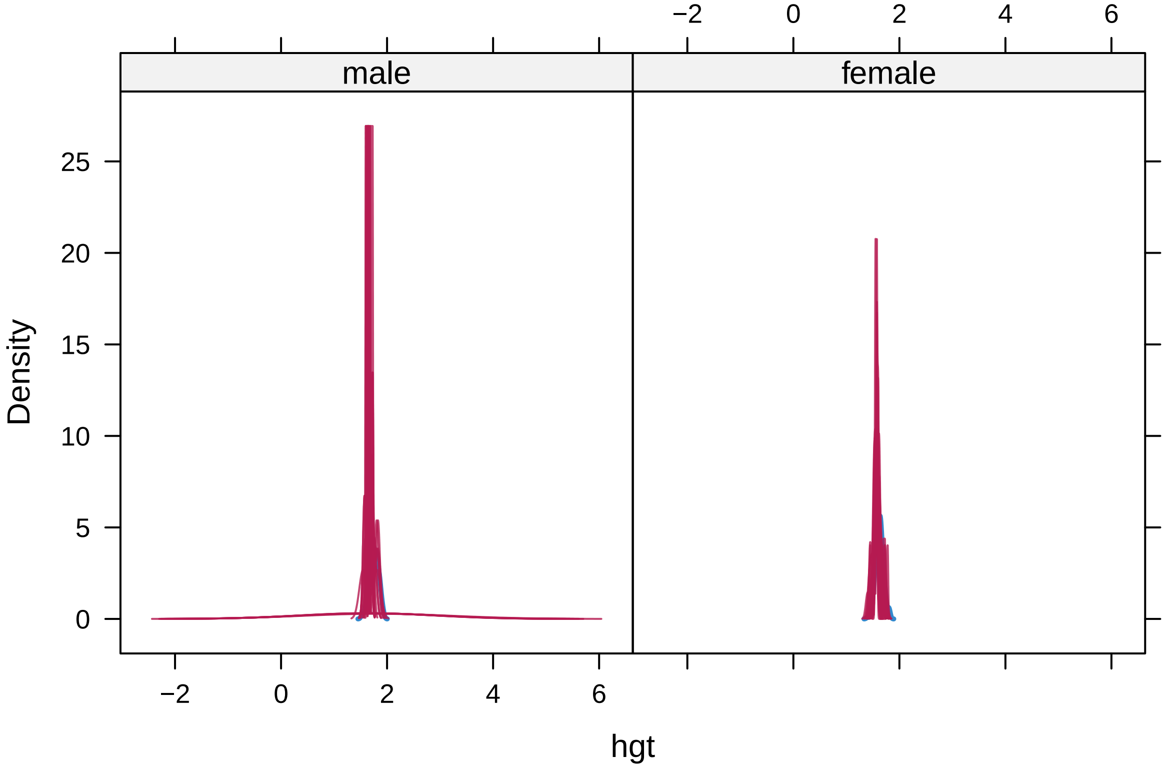


Most fitting distributions are good, but we could still notice that the `hgt` and the `SBP` terms are not satisfying enough. Although there is nothing we could do further to modify these patterns, we could still inspect whether such differences between observed and imputed distributions could be explained by other variables. Here we check the `SBP` conditional on the `hypten` and `gender` as well as the `hgt` conditional on `gender` respectively.

```
densityplot(imps5, ~SBP|hypten + gender)
```



```
densityplot(imps5, ~hgt|gender)
```

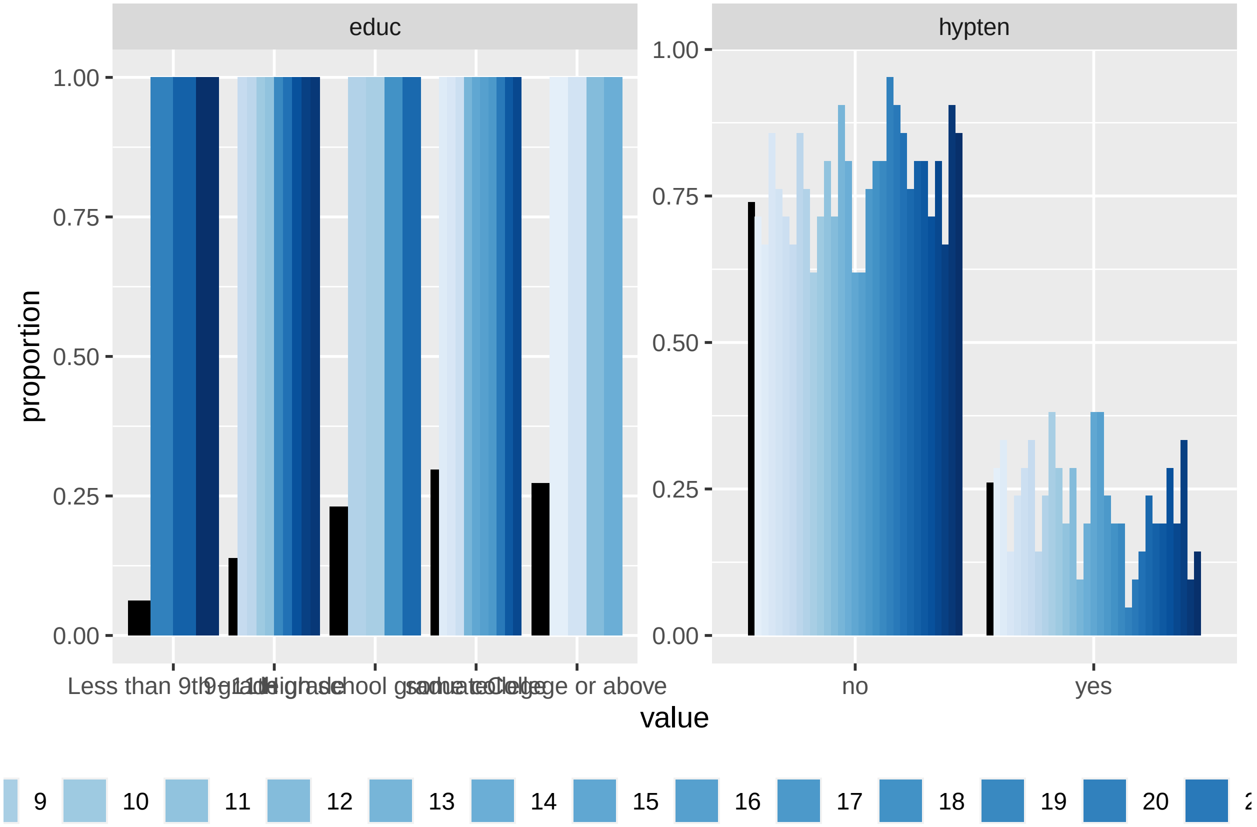


From the comparison, the hypertensive status and gender can really explain some of the differences between the observed and imputed distributions of `SBP` to some extent. Gender can help the explanation in the case of `hgt`, too.

As for the binary/categorical variables (factors), we compare the proportion of values in each category.

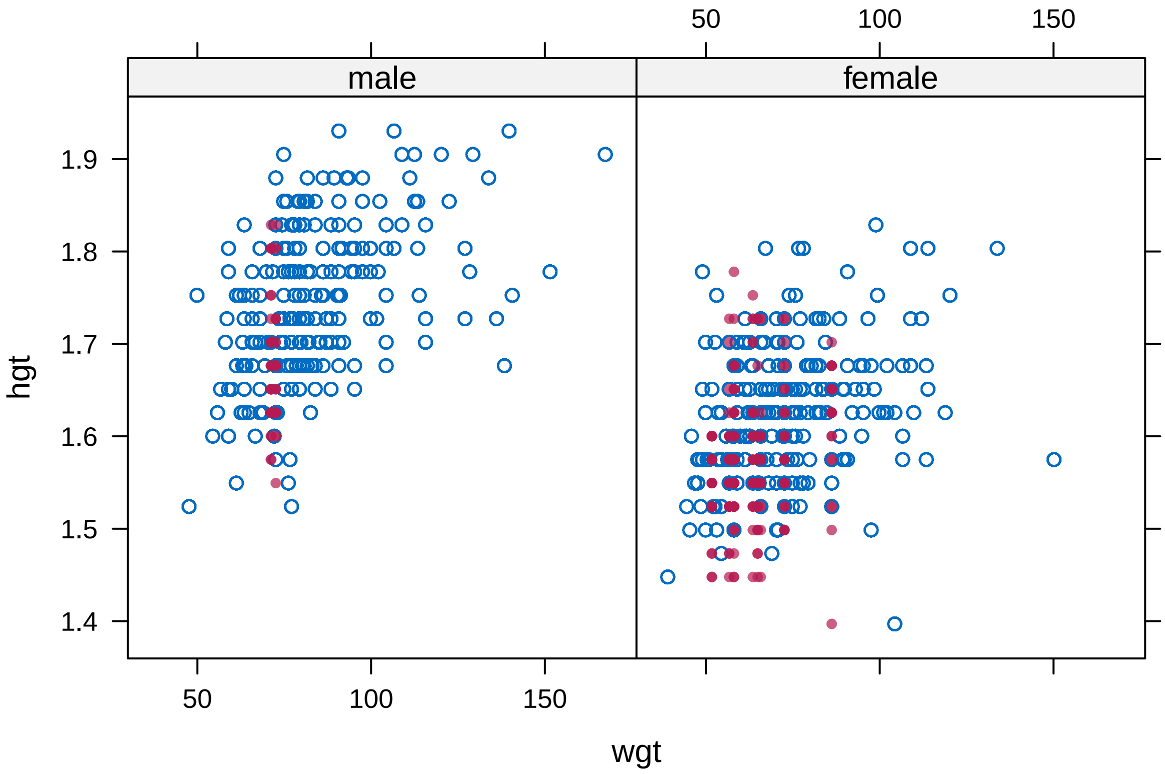
```
propplot(imps5)
```

```
## Warning: attributes are not identical across measure
## variables; they will be dropped
```



Still, for the `educ` term, since there is only one missing value, the pattern seems strange but indeed reasonable. Also, even if there is any discrepancy, due to the fact that the whole dataset has 500 rows, the effect is really slight and would not be too problematic. We can also visualize the imputed and observed values for pairs of variables. Here we choose to view the relationship between height `hgt` and weight `wgt` conditional on `gender`.

```
xyplot(imps5, hgt ~ wgt | gender, pch = c(1, 20))
```



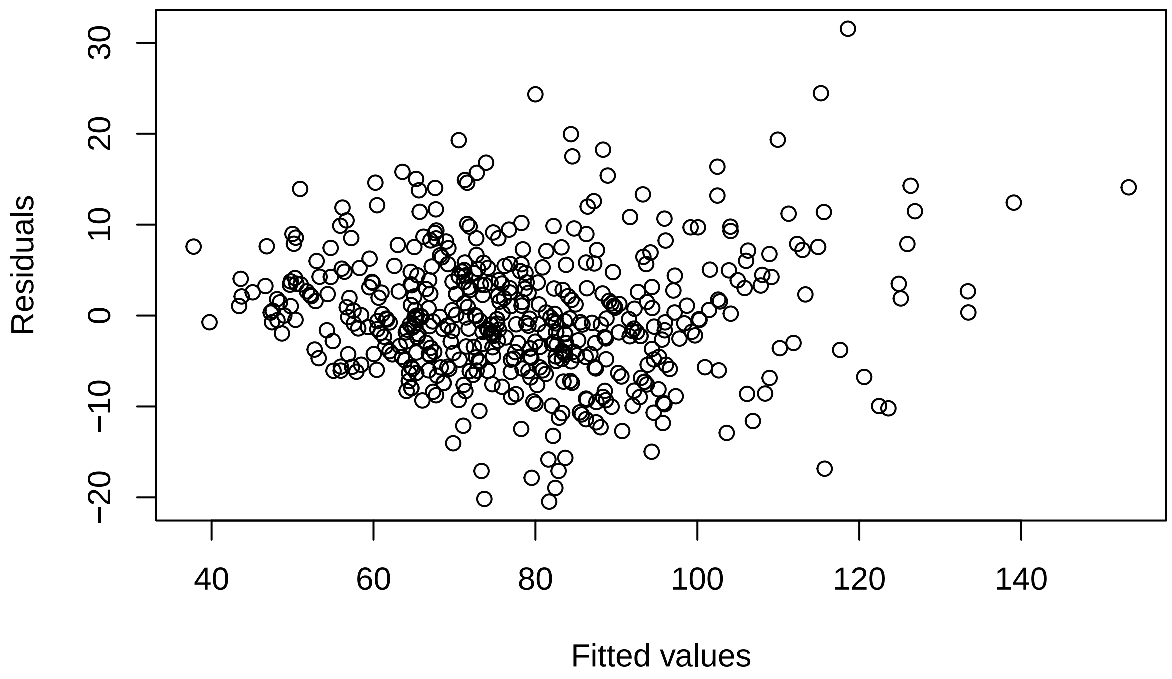
Note that `wgt` and `gender` are both fully observed, while only `hgt` has missing values. Now we almost confirm that the imputation step is successful and reasonable, so that we proceed to the analysis of the imputed data and fit the interested regreesion model. For the fitted model, firstly we look at its summary.

```
fits5 <- with(imps5, lm(wgt ~ gender + age + hgt + WC))
summary(fits5$analyses[[1]])
```

```
##
## Call:
## lm(formula = wgt ~ gender + age + hgt + WC)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.4638  -4.5537  -0.4955   3.8854  31.5403
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -100.51035    7.50652 -13.390  < 2e-16 ***
## genderfemale   -1.26815    0.81952  -1.547    0.122
## age            -0.15827    0.02085  -7.590  1.6e-13 ***
## hgt            52.15392    4.29615  12.140  < 2e-16 ***
## WC              1.02795    0.02213  46.452  < 2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.179 on 495 degrees of freedom
## Multiple R-squared:  0.8575, Adjusted R-squared:  0.8563
## F-statistic: 744.6 on 4 and 495 DF,  p-value: < 2.2e-16
```

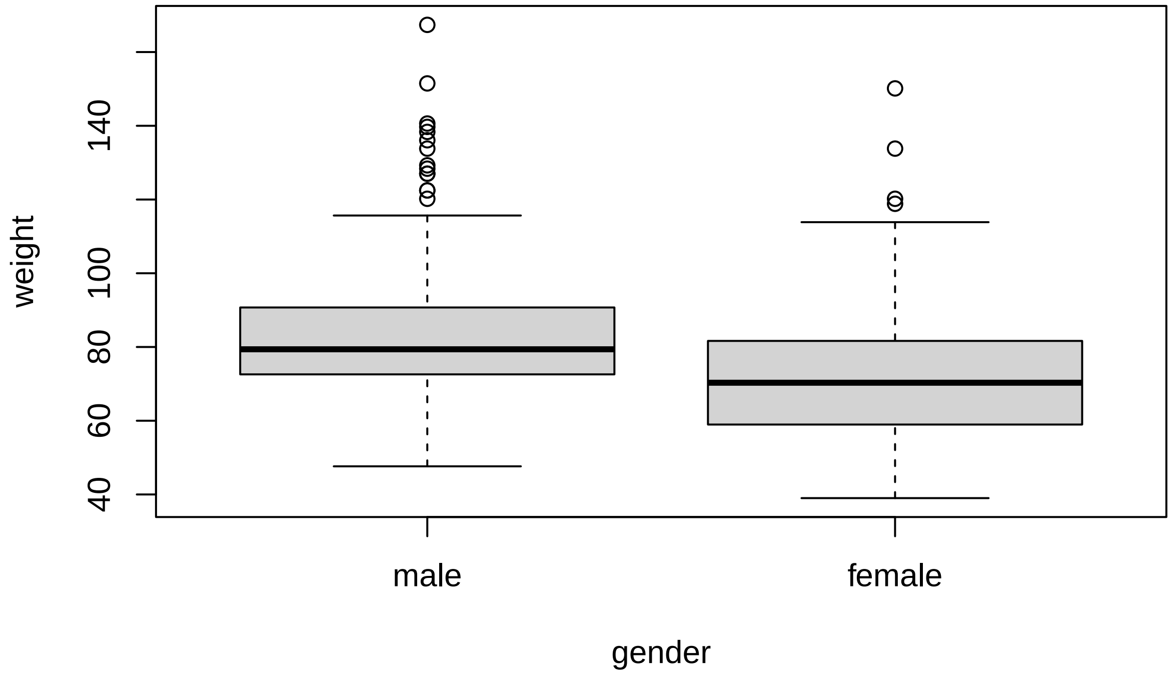
From the fitting summary, it seems that the `gender` is not significant enough, indicating that it does not offer enough information during the regression. With the fitting result, we can also check the fitted values versus residuals plot.

```
comp5 <- complete(imps5, 1)
plot(fits5$analyses[[1]]$fitted.values, residuals(fits5$analyses[[1]]),
     xlab = "Fitted values", ylab = "Residuals")
```

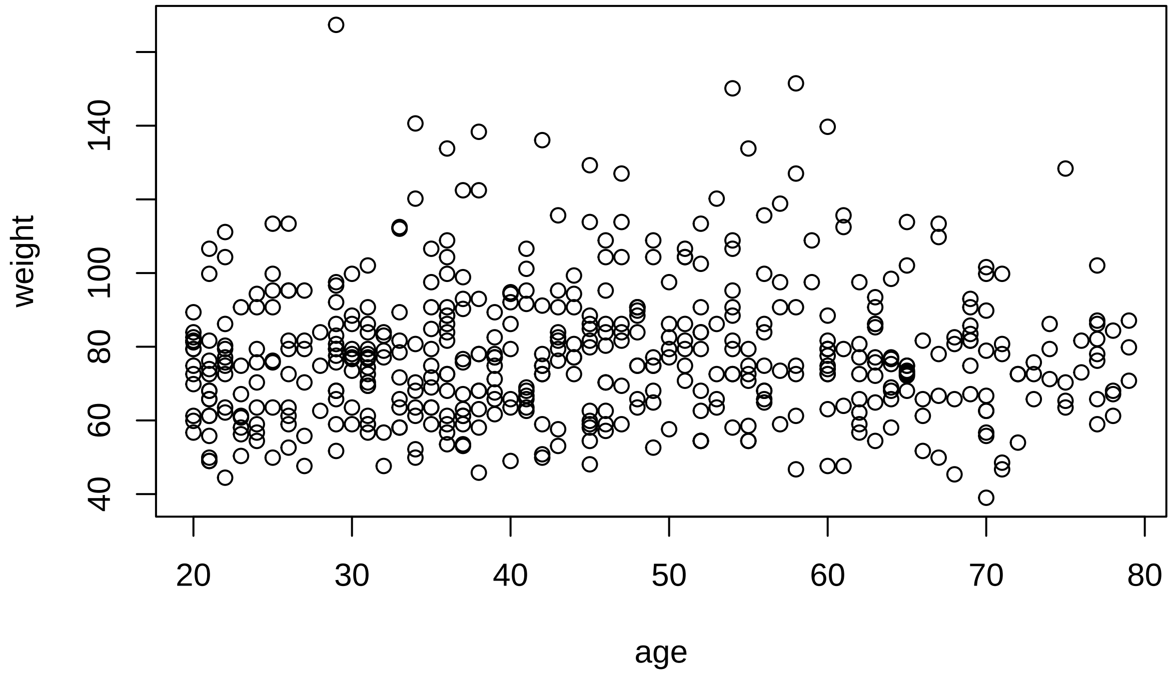


Seems good. And we can also plot the response variable weight `wgt` against other variables.

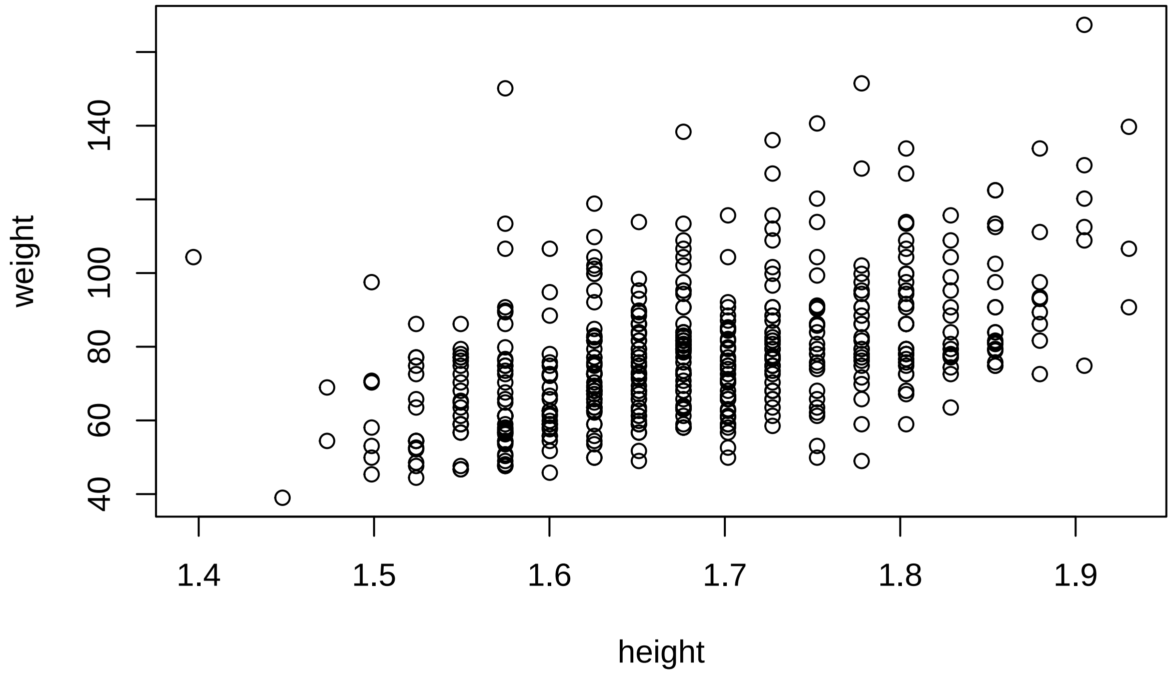
```
boxplot(comp5$wgt ~ comp5$gender, xlab = "gender", ylab = "weight")
```



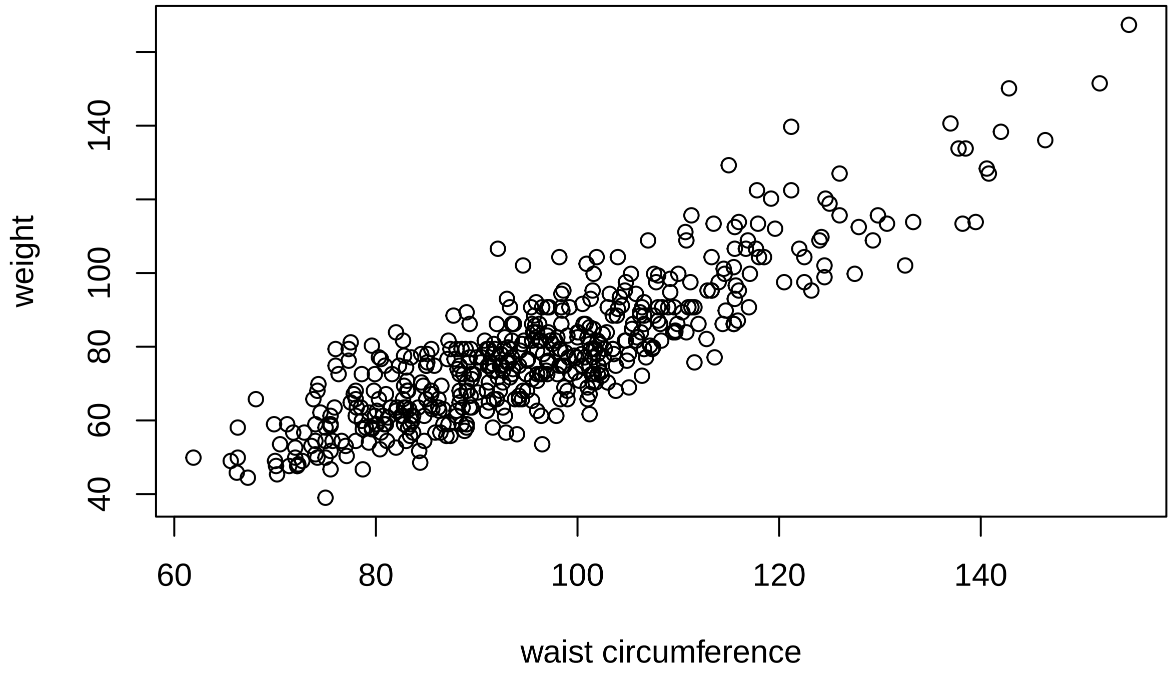
```
plot(comp5$wgt ~ comp5$age, xlab = "age", ylab = "weight")
```



```
plot(comp5$wgt ~ comp5$hgt, xlab = "height", ylab = "weight")
```

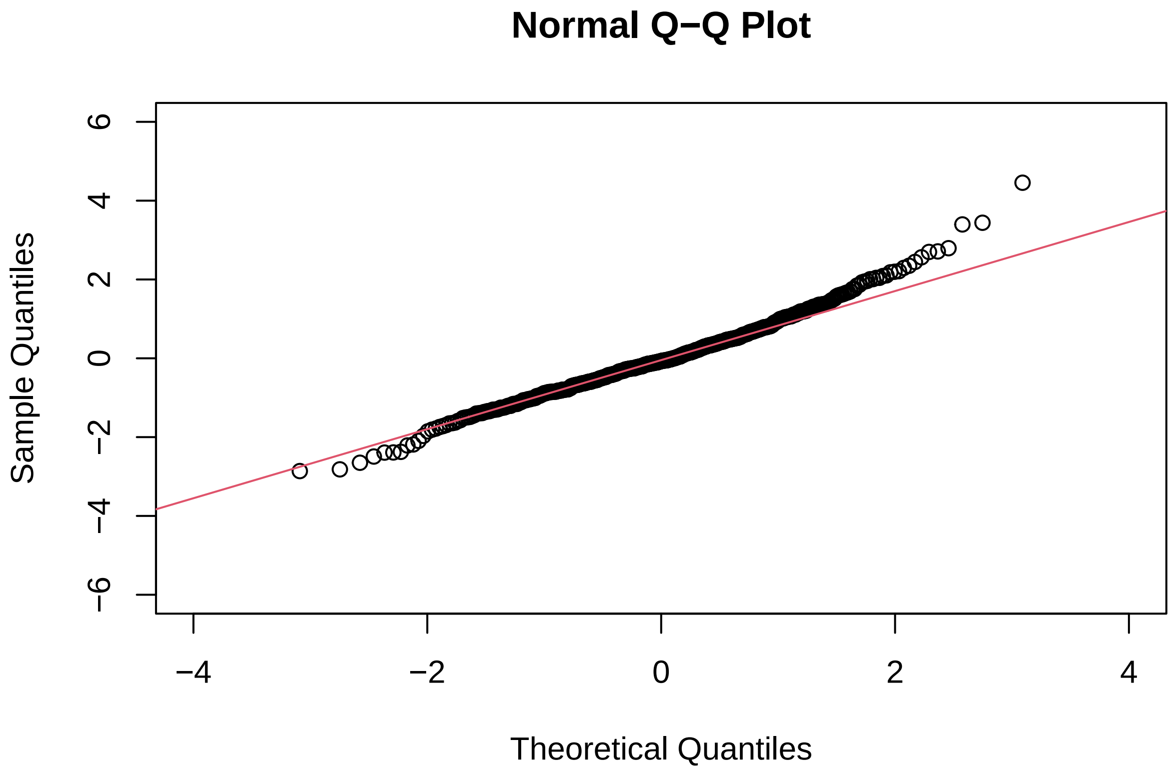


```
plot(comp5$wgt ~ comp5$WC, xlab = "waist circumference", ylab = "weight")
```



With the QQplot, nothing looks suspicious, too.

```
qqnorm(rstandard(fits5$analyses[[1]]), xlim = c(-4, 4), ylim = c(-6, 6))
qqline(rstandard(fits5$analyses[[1]]), col = 2)
```



Now we pool the results and check the summary.

```
pooled_ests5 <- pool(fits5)
summary(pooled_ests5, conf.int = TRUE)
```

```
##           term     estimate  std.error  statistic       df
## 1  (Intercept) -100.8520702 7.67289217 -13.143945 449.6454
## 2 genderfemale   -1.3850796 0.83448095  -1.659810 469.9080
## 3          age   -0.1576829 0.02141424  -7.363458 451.6777
## 4          hgt   52.4292200 4.39636603  11.925581 444.3719
## 5           WC    1.0260613 0.02232811  45.953795 481.3369
##        p.value        2.5 %      97.5 %
## 1 0.000000e+00 -115.9312510 -85.7728894
## 2 9.761988e-02   -3.0248557   0.2546965
## 3 8.562040e-13   -0.1997668  -0.1155990
## 4 0.000000e+00   43.7889680  61.0694720
## 5 0.000000e+00    0.9821887   1.0699340
```

Now evaluate the model fit (which could also be used in comparison of different models' performance).

```
pool.r.squared(pooled_ests5, adjusted = TRUE)
```

```
##                est     lo 95     hi 95        fmi
## adj R^2 0.8559941 0.8304596 0.8779613 0.02126541
```

Re-consider the significant problem of `gender` term, we choose a multivariate Wald test and a likelihood-ratio test statistic, respectively.

```
fit_no_gender <- with(imps5, lm(wgt ~ age + hgt + WC))
D1(fits5, fit_no_gender)
```

```
##    test statistic df1     df2 dfcom    p.value        riv
##  1 ~~ 2  2.754968   1 481.979   495 0.09760318 0.03241167
```

```
D3(fits5, fit_no_gender)
```

```
##    test statistic df1      df2 dfcom    p.value        riv
##  1 ~~ 2  2.770247   1 20233.53   495 0.09604684 0.03392227
```

Both results indicate that at a 5% level of confidence, the term `gender` is not significant. By deleting this unsignificant term, the corresponding result is shown as below.

```
pooled_ests5_2 <- pool(fit_no_gender)
summary(pooled_ests5_2, conf.int = TRUE)
```

```
##          term     estimate  std.error  statistic       df
## 1 (Intercept) -108.9902721 5.92539388 -18.393760 429.4458
## 2         age   -0.1541883 0.02135993  -7.218578 451.4300
## 3         hgt   56.8772606 3.49544832  16.271807 425.2636
## 4          WC    1.0239541 0.02233220  45.851012 482.4802
##        p.value        2.5 %      97.5 %
## 1 0.000000e+00 -120.6366536 -97.3438906
## 2 2.242428e-12   -0.1961656  -0.1122111
## 3 0.000000e+00   50.0067542  63.7477669
## 4 0.000000e+00    0.9800737   1.0678345
```

Finally, we adopt the adjusted-R^2, and compare the two models.

```
pool.r.squared(pooled_ests5, adjusted = TRUE); pool.r.squared(pooled_ests5_2, adjusted = TRUE)
```

```
##                est     lo 95     hi 95        fmi
## adj R^2 0.8559941 0.8304596 0.8779613 0.02126541
```

```
##                est     lo 95     hi 95        fmi
## adj R^2 0.8554496 0.8298317 0.8774914 0.02102272
```

Finally, the two adjusted-R^2s are really close, while the performance of the first one (original one, without deleting the unsignificant term) is slightli better than the modified one. However, in practice, I personally would recommend the second one, with a neater model structure (furthermore, information criteria such as AIC or BIC could also be applied to compare these two models).