Assignment 3
Please find the code on this link from GitHub (https://github.com/alexayu1204/IDA_assignment3)

1.
a.
There are 48% cases which not complete.

b.
λ value show how much missing data affect each variable. From code results, we find that highest λ value for age2 is 0.6107, and lowest for hyp2 is 0.2635. So, we can say age is most impacted by non-response.

c.
When we look at results, we see that parameter with biggest impact change based on seed value. For example, with seed 2, hyp2 has highest λ value (0.516). But when seed values 3, 4, 5, or 6, age has highest λ value again. This instability maybe because 48% cases not complete.

d.
From results, we see when M is bigger, λ change more smoothly. We think age is most important parameter because it always has highest λ. So, we prefer M=100 over M=5. We believe age is most important parameter because age always has highest λ. We prefer M=100 over M=5 because of this.

2.
Stochastic regression imputation gives empirical coverage probability 0.88 for 95% confidence intervals of β1. But, bootstrap-based method has higher probability, 0.95.
For bootstrap method, we find true value of β1 is inside interval 95% of the time. But for SRI method, it happens only 88% of the time. This difference is because SRI method does not think about uncertainty when making imputed values. It treats them like real observed data, not missing values that have been imputed. Because of this, confidence intervals for β1 in SRI method become too narrow and too confident, making them less likely to have true value inside.

3.
For strategy one,

We have the imputed dataset as: $Y_{imp,i} = X_{imp} * \beta_i, for\ i\ =\ 1 \ldots M$
And we can pool the predict values using Rubin's rule, as following:

$$Y_{imp,i} = \frac{1}{M}\sum_{i=1}^{M} Y_{imp,i}$$

For strategy 2,
For each imputed dataset, we fit the regression model and obtain the regression coefficients. We average these coefficients using Rubin's rule. We will have:

$$\beta_{pool} = \frac{1}{M}\sum_{i=1}^{M}\beta_i$$

Thus

$$Y_{imp,val} = X_{imp} * \beta_{pool} = \beta_{pool} * \frac{1}{M}\sum_{i=1}^{M}\beta_i$$

Therefore, we have

$$(X_{imp} * \frac{1}{M}\sum_{i=1}^{M}\beta_i) = \frac{1}{M}\sum_{i=1}^{M}(X_{imp} * \beta_i) = \frac{1}{M}\sum_{i=1}^{M}Y_{imp,i}$$

Therefore strategy 1 and 2, they are mathematically equivalent.

4.
a.
β1 = 1.41; 95% CI = [1.22, 1.60]
β2 = 1.97; 95% CI = [1.86, 2.07]
β3 = 0.75; 95% CI = [0.642, 0.868]
We applied the impute, then transform technique for imputation in this portion of our solution. We opted to explicitly impute the x1 and y variables with the MICE method, and then utilized the imputed x1 values for passive imputation of the interaction term, x1x2, within our substantive model.
Setting M to 50, seed to 1, and maxit to 5, we calculated the estimates and 95% confidence intervals (CIs) for β1, β2, and β3

b.
β1 = 1.19; 95% CI = [1.0035, 1.38]
β2 = 2.00; 95% CI = [1.90, 2.09]
β3 = 0.874; 95% CI = [0.762, 0.987]
We added interaction term to dataset, z = x1x2, and used passive imputation for missing values, keeping deterministic relationship. With M=50, seed=1, maxit=5, we got these estimates and 95% CIs for β1, β2, and β3
With this imputation method, we see improved accuracy for all three parameters' estimates. While 95% CIs for β1 and β3 still don't have true values of these parameters, they are closer

than method in Q4a. Passive imputation reduced bias in estimates from substantive model but did not completely solve it.

c.
β1 = 1.0039; 95% CI = [0.841, 1.17]
β2 = 2.03; 95% CI = [1.94, 2.11]
β3 = 1.02; 95% CI = [0.930, 1.105]
Doing same process as in 4a and 4b with M=50, seed=1, maxit=5, we got these estimates and 95% CIs for β1, β2, and β3
Now, we treat interaction term as just another variable, estimates are accurate to at least 1 decimal place and all three 95% CIs contain true parameter values. This is big improvement compared to results in 4a and 4b.

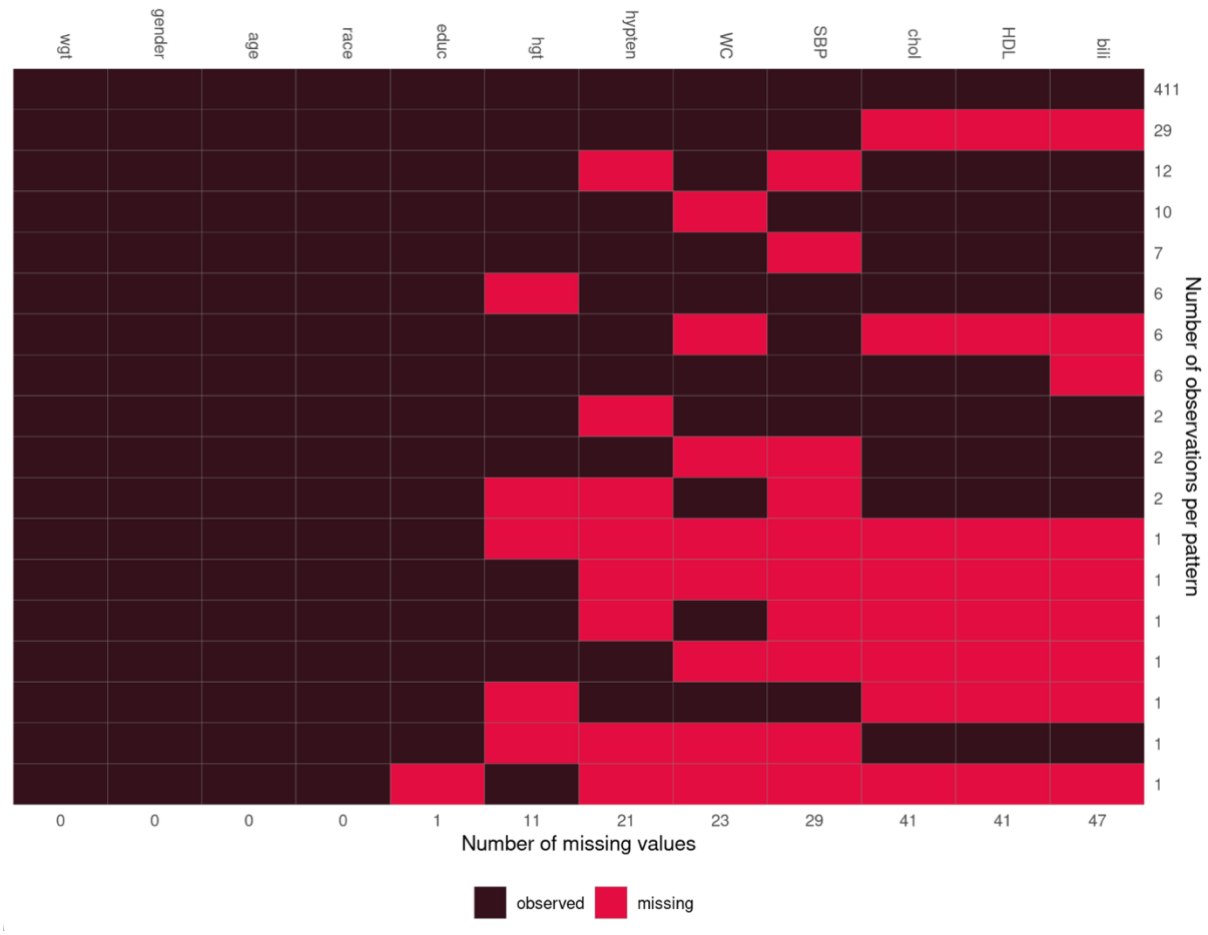
d.
The key problem when using this method to get unbiased estimates of parameters in a linear regression model is that the deterministic connection between interaction term, $z = x_1x_2$, is not maintained.
To be more specific, imputing missing values of z without considering x_1 causes the dependency between these variables to disappear. Hence x_1x_2 is not equivalent to the product of x_1 and x_2

5.

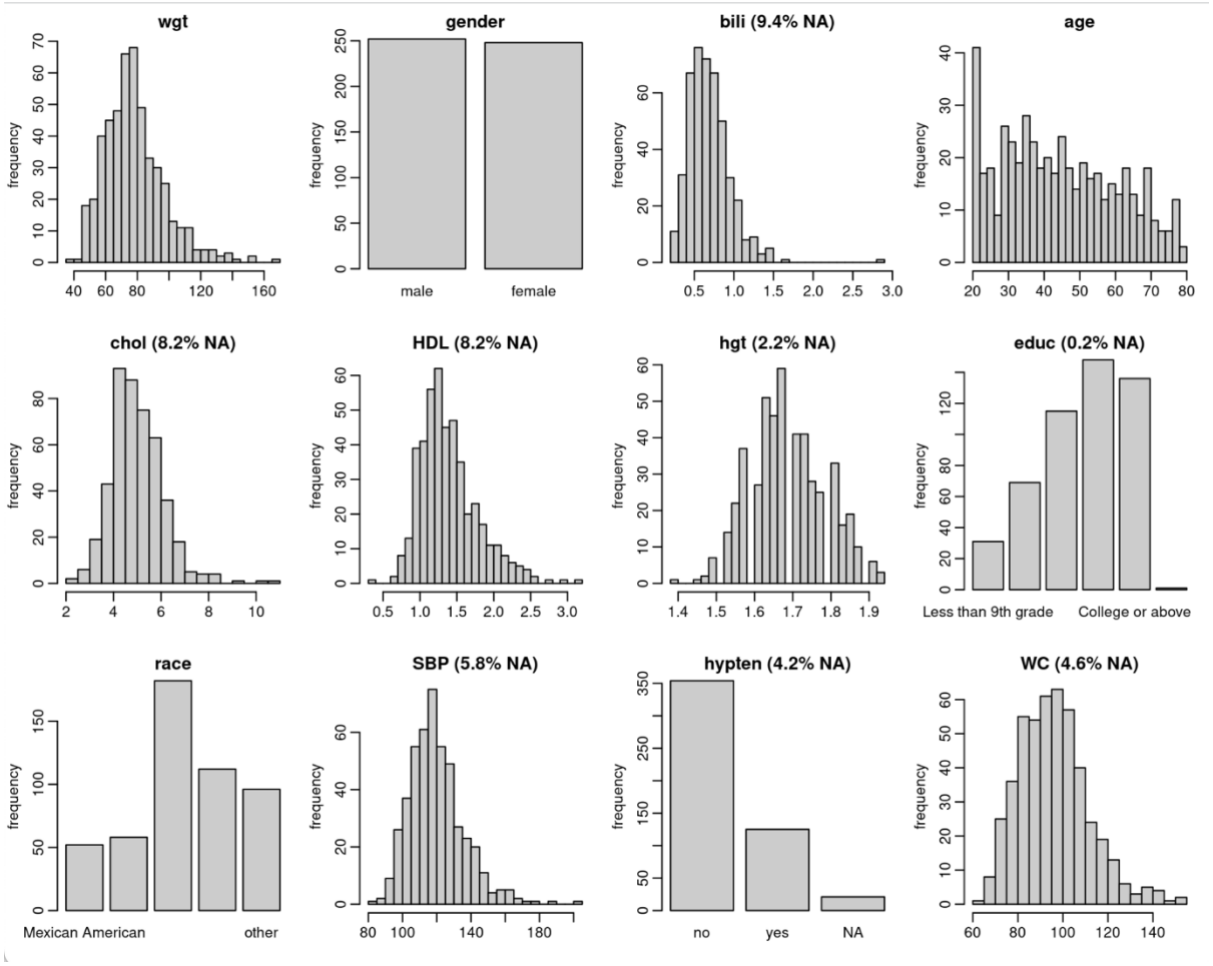
```
> load("NHANES2.Rdata")
> dim(NHANES2)
[1] 500  12
> str(NHANES2)
'data.frame':   500 obs. of  12 variables:
 $ wgt   : num  78 78 75.3 90.7 112 ...
 $ gender: Factor w/ 2 levels "male","female": 1 1 2 1 2 1 2 2 1 1 ...
 $ bili  : num  1.1 0.7 0.5 0.8 0.6 0.7 1.1 0.8 0.8 0.5 ...
 $ age   : num  67 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 ...
```

```
> summary(NHANES2)
      wgt            gender         bili             age             chol            HDL              hgt        
 Min.   : 39.01   male  :252   Min.   :0.2000   Min.   :20.00   Min.   : 2.07   Min.   :0.360   Min.   :1.397  
 1st Qu.: 65.20   female:248   1st Qu.:0.6000   1st Qu.:31.00   1st Qu.: 4.27   1st Qu.:1.110   1st Qu.:1.626  
 Median : 76.20                Median :0.7000   Median :43.00   Median : 4.86   Median :1.320   Median :1.676  
 Mean   : 78.25                Mean   :0.7404   Mean   :45.02   Mean   : 5.00   Mean   :1.395   Mean   :1.687  
 3rd Qu.: 86.41                3rd Qu.:0.9000   3rd Qu.:58.00   3rd Qu.: 5.64   3rd Qu.:1.590   3rd Qu.:1.753  
 Max.   :167.38                Max.   :2.9000   Max.   :79.00   Max.   :10.68   Max.   :3.130   Max.   :1.930  
                               NA's   :47                       NA's   :41      NA's   :41      NA's   :11     
                 educ                        race          SBP          hypten          WC        
 Less than 9th grade : 31   Mexican American  : 52   Min.   : 81.33   no  :354   Min.   : 61.90  
 9-11th grade        : 69   Other Hispanic    : 58   1st Qu.:109.00   yes :125   1st Qu.: 84.80  
 High school graduate:115   Non-Hispanic White:182   Median :118.67   NA's: 21   Median : 95.00  
 some college        :148   Non-Hispanic Black:112   Mean   :120.05              Mean   : 96.07  
 College or above    :136   other             : 96   3rd Qu.:128.67              3rd Qu.:104.80  
 NA's                :  1                            Max.   :202.00              Max.   :154.70  
                                                     NA's   :29                  NA's   :23      
```

We can first get some basic information for the data
The, we take use of jointAI package to get some insights on missing data



there are 411 data points with complete information for all 12 factors. Additionally, there are 6 instances where only the height in meters (hgt) is not available, and so on. For more verification, we can examine how the various factors relate to each other.Next, we check if the assumption of normality is generally satisfied. We employ the JointAI package to see the distribution of the observed portions of the incomplete factors.

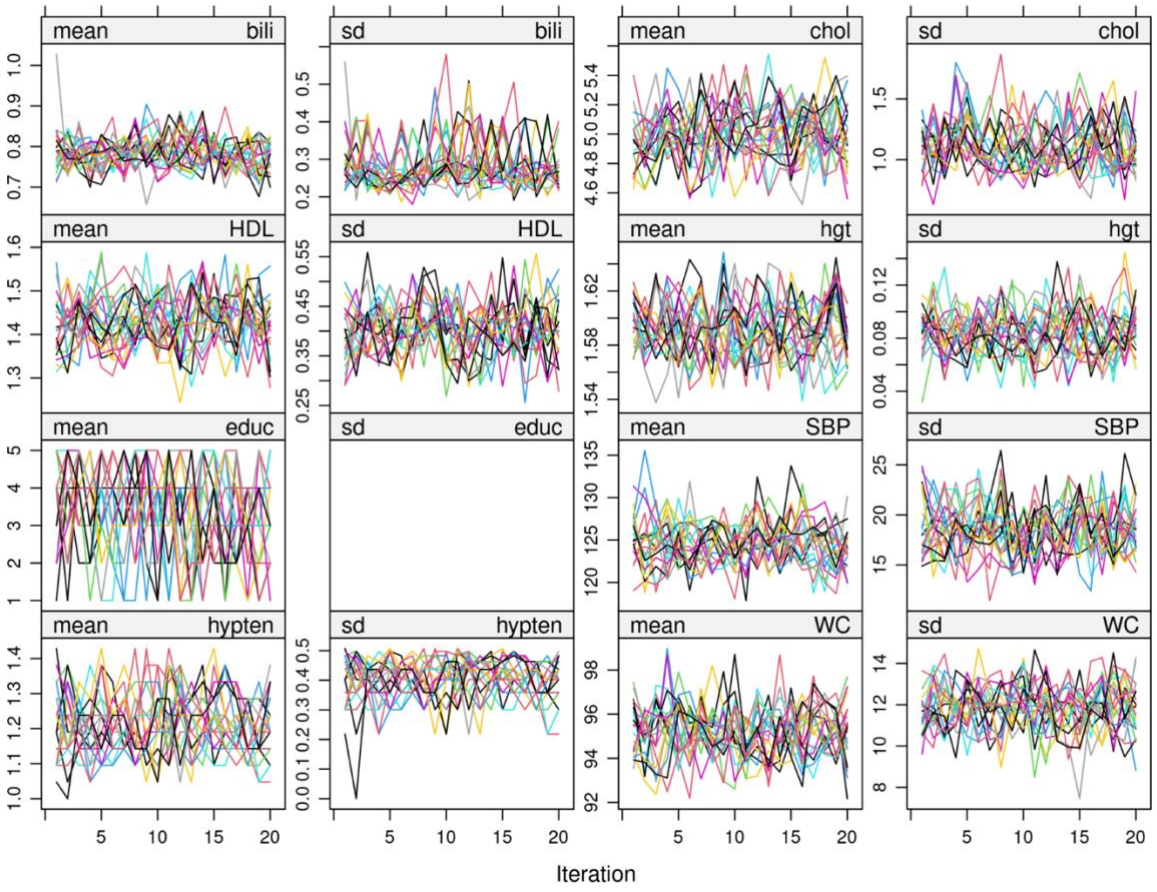


Having inspected the data, we are ready to start our imputation procedure.

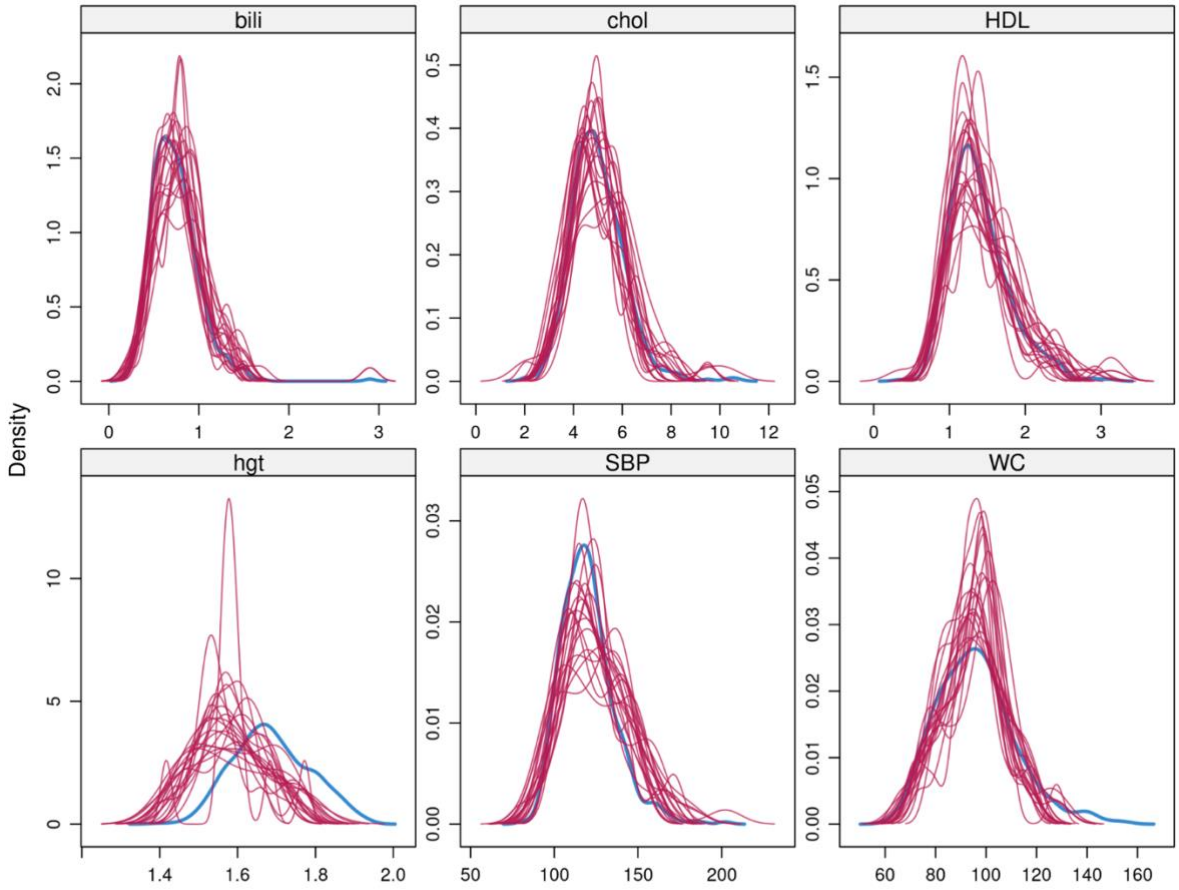
```
> imp0 <- mice(NHANES2, maxit = 0)
> imp0
Class: mids
Number of multiple imputations:  5
Imputation methods:
     wgt   gender     bili      age     chol      HDL      hgt     educ     race      SBP   hypten       WC
      ""       ""    "pmm"       ""    "pmm"    "pmm"    "pmm"   "polr"       ""    "pmm" "logreg"    "pmm"
PredictorMatrix:
       wgt gender bili age chol HDL hgt educ race SBP hypten WC
wgt      0      1    1   1    1   1   1    1    1   1      1  1
gender   1      0    1   1    1   1   1    1    1   1      1  1
bili     1      1    0   1    1   1   1    1    1   1      1  1
age      1      1    1   0    1   1   1    1    1   1      1  1
chol     1      1    1   1    0   1   1    1    1   1      1  1
HDL      1      1    1   1    1   0   1    1    1   1      1  1
>
> meth <- imp0$method
> meth["hgt"] <- "norm"
> meth
     wgt   gender     bili      age     chol      HDL      hgt     educ     race      SBP   hypten       WC
      ""       ""    "pmm"       ""    "pmm"    "pmm"   "norm"   "polr"       ""    "pmm" "logreg"    "pmm"
```

We set in the code that all estimates of hgt outside the interval (0, 2.5) will be set to these limit values since we don't want height to me negative and then I will set maxit = 20 and M = 20 to perform the next imputation.

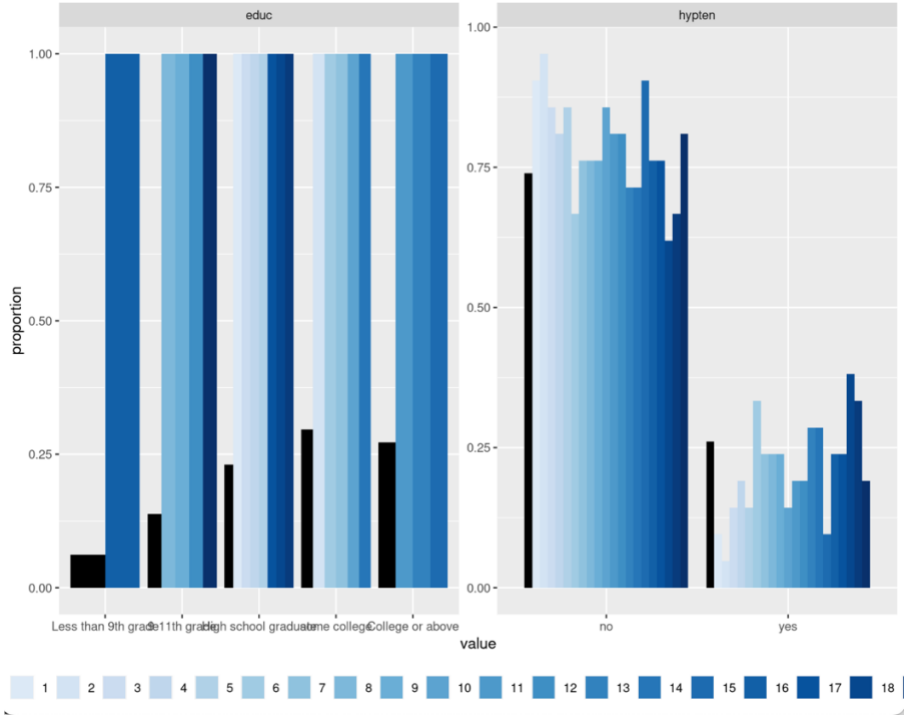
We should check if mice actually has converged by plotting our object and visualize the traceplots.



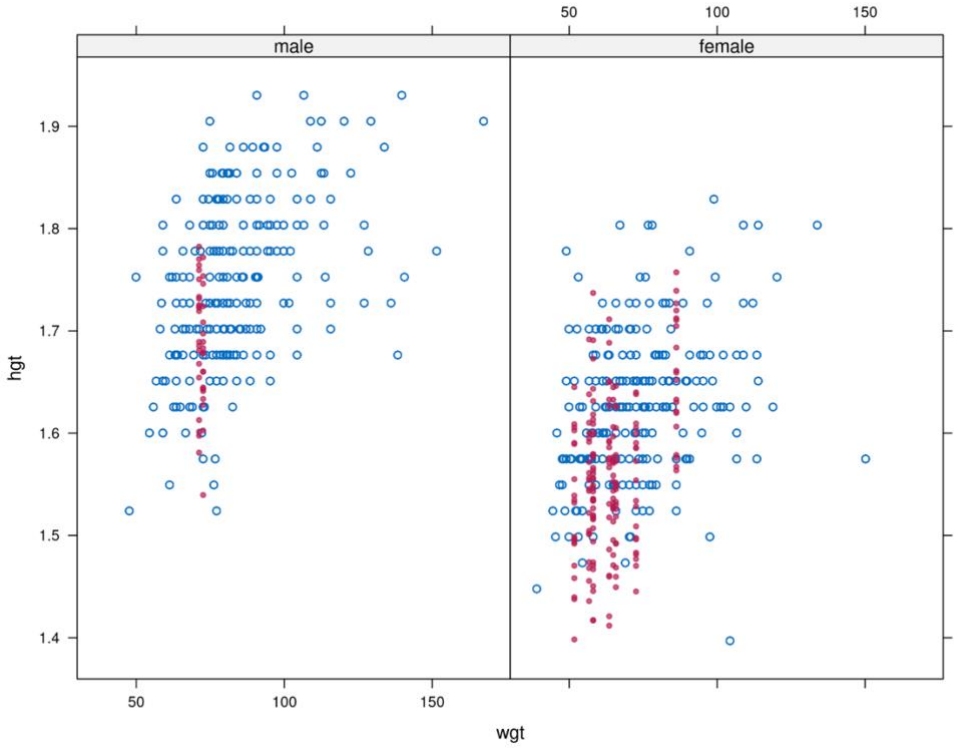
Now, understanding that the iterative method seems to converge for all factors that underwent imputation, we can contrast the distribution of these imputed values with the distribution of the observed values. We initiate this process for continuous factors.



In relation to binary or categorical factors, we can examine the proportion of values in each category. While mice doesn't offer a function for this, there is an effective one called propplot, developed by Nicole Erler and accessible on her GitHub.

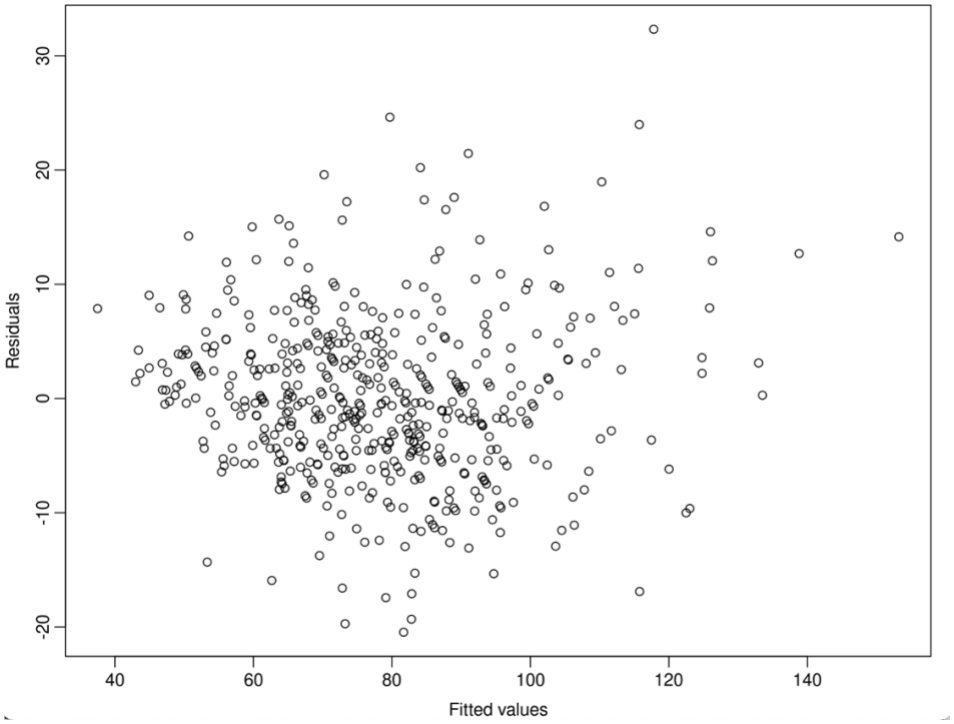


Based on the above image, we can notice the distinctions between the observed and estimated data distributions for the factors. The xyplot() function enables the visualization of scatterplots for imputed and observed values in variable pairs.

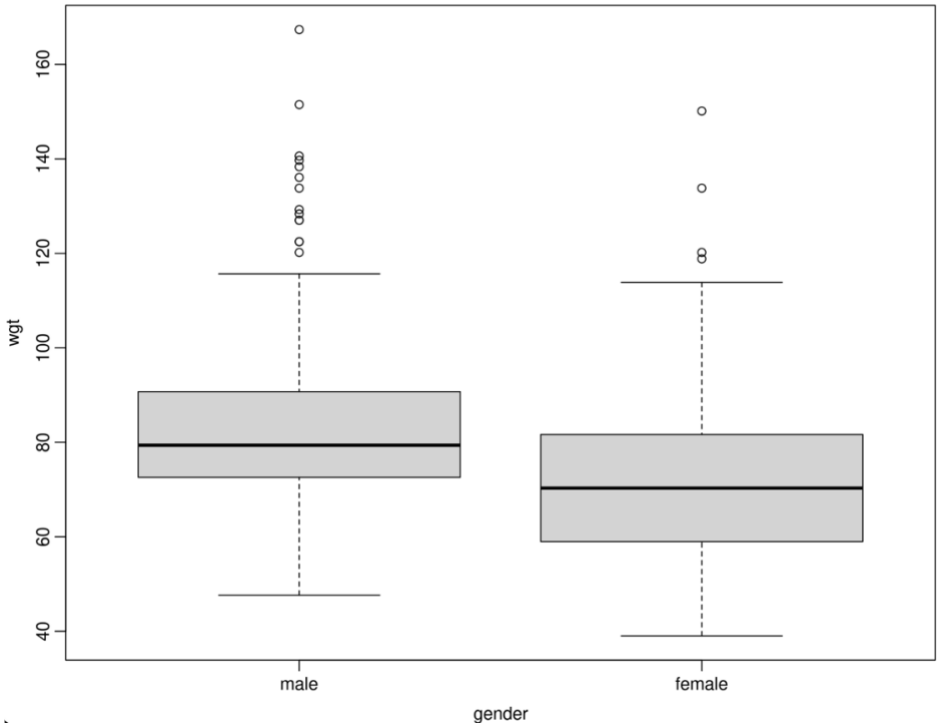


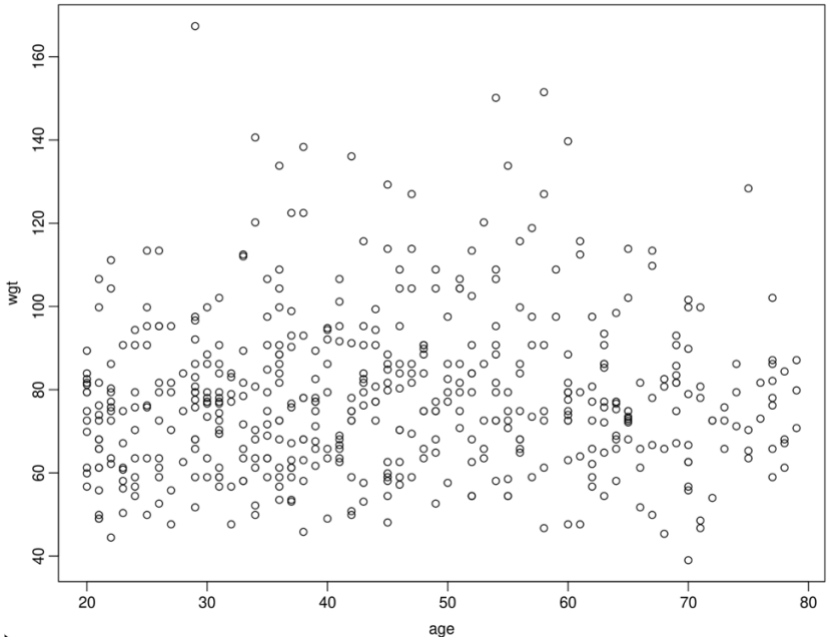
Having confirmed that our imputation step was successful, we can proceed to the analysis of the imputed data.

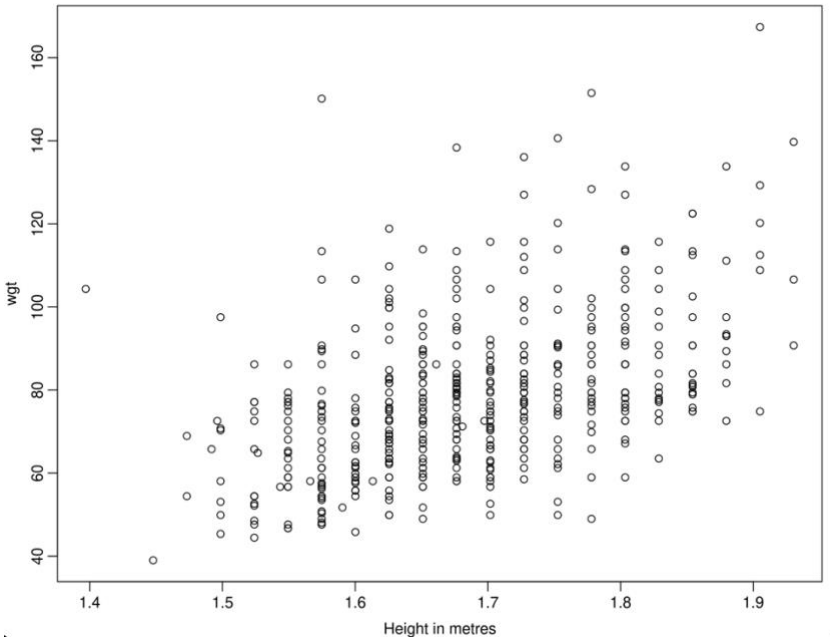
```
Call:
lm(formula = wgt ~ gender + age + hgt + WC)

Residuals:
    Min      1Q  Median      3Q     Max
-20.463  -4.585  -0.385   4.162  32.323

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -104.98644    7.55808 -13.891  < 2e-16 ***
genderfemale   -1.23947    0.83155  -1.491    0.137
age            -0.15348    0.02106  -7.288 1.25e-12 ***
hgt            55.12737    4.30910  12.793  < 2e-16 ***
WC              1.01901    0.02225  45.792  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.259 on 495 degrees of freedom
Multiple R-squared:  0.8543,    Adjusted R-squared:  0.8531
F-statistic: 725.6 on 4 and 495 DF,  p-value: < 2.2e-16
```

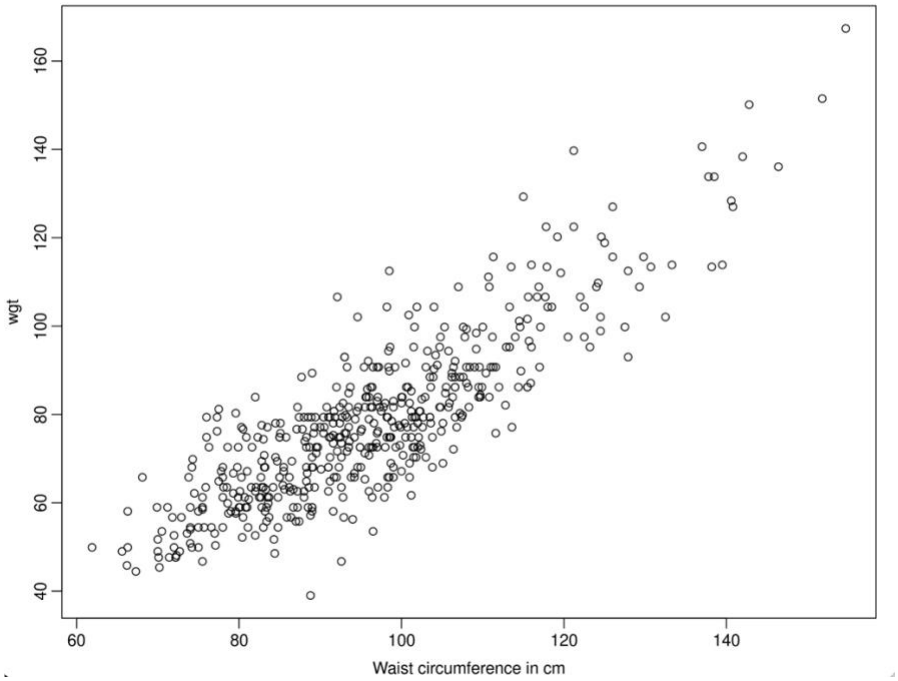


We can see from the residual plot, data clustered in the left down area of the graph. Furthermore, we can plot wgt against every feature

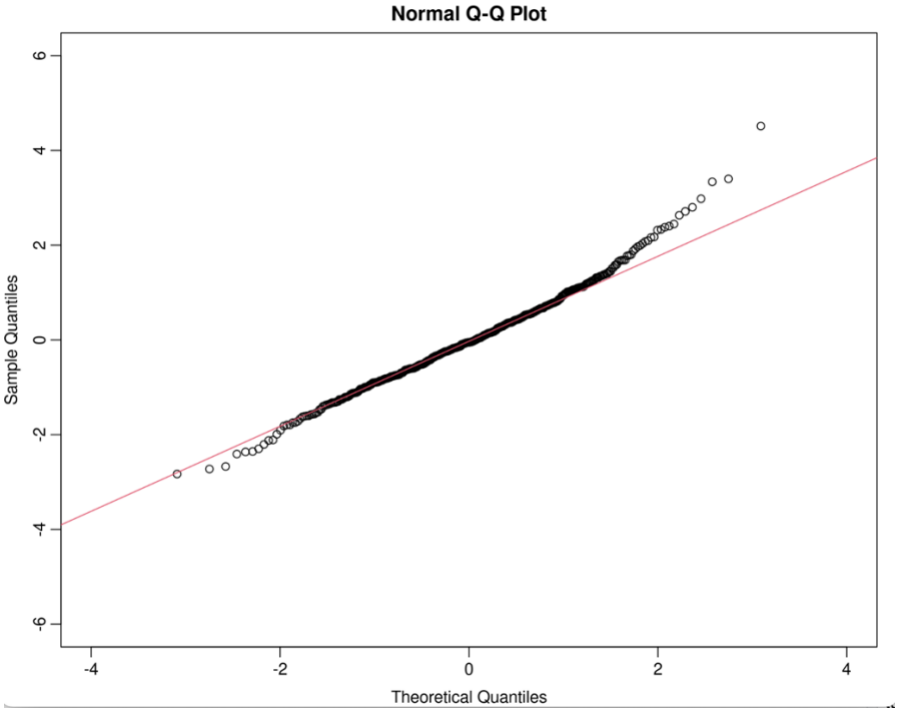



wgt
age
40
60
80
100
120
140
160
20
30
40
50
60
70
80


wgt
Height in metres
40
60
80
100
120
140
160
1.4
1.5
1.6
1.7
1.8
1.9



We also draw a normal QQ plot, looks fine



Then we can summary the results:

```
> summary(pooled_ests, conf.int = TRUE)
          term    estimate  std.error  statistic       df       p.value        2.5 %      97.5 %
1  (Intercept) -100.960341 7.70806594 -13.098012 410.3546  5.404651e-33 -116.1125628 -85.8081195
2 genderfemale   -1.323852 0.82988381  -1.595226 476.7304  1.113244e-01   -2.9545344   0.3068302
3          age   -0.155543 0.02158669  -7.205505 414.3349  2.753943e-12   -0.1979761  -0.1131099
4          hgt   52.498011 4.40996593  11.904403 407.6217  3.004078e-28   43.8288964  61.1671255
5           WC    1.024904 0.02238150  45.792460 471.9279 9.340894e-176    0.9809240   1.0688835
```

Furthermore, we can use mice to evaluate the model fit.
As we can see from the following results, the Wald test statistic displayed no significance, with a p-value greater than 0.05. Nevertheless, in the second, third, and fourth cases, the Wald test statistic demonstrated significance, suggesting that age, hgt, and WC must be part of the model. Consequently, we determine that age, height, and WC possess a more substantial influence compared to gender.

```
> fit_no_gender <- with(imp, lm(wgt ~ age + hgt + WC))
> D1(fit, fit_no_gender)
   test statistic df1      df2 dfcom  p.value        riv
 1 ~~ 2  2.544746   1 483.5937   495 0.111315 0.02206217
>
> fit_no_age<- with(imp, lm(wgt ~ gender + hgt + WC))
> D1(fit, fit_no_age)
   test statistic df1      df2 dfcom      p.value        riv
 1 ~~ 2  51.91931   1 415.5833   495 2.742024e-12 0.07220702
>
> fit_no_hgt<- with(imp, lm(wgt ~ age+ gender + WC))
> D1(fit, fit_no_hgt)
   test statistic df1      df2 dfcom      p.value        riv
 1 ~~ 2  141.7148   1 407.5967   495 3.005617e-28 0.07696442
>
> fit_no_WC<- with(imp, lm(wgt ~ age + gender + hgt))
> D1(fit, fit_no_WC)
   test statistic df1     df2 dfcom       p.value        riv
 1 ~~ 2  2096.949   1 479.315   495 3.693172e-177 0.02688739
```