Assignment 3

Zekai Li S2040608

Question 1

(*a*) My solution

```r
print(c("The propotion of incomplete cases is",
        1-nrow(cc(nhanes))/nrow(nhanes)),quote=FALSE)
```

```
## [1] The propotion of incomplete cases is 0.48
```

The percentage of incomplete cases is 48%.

(*b*) My solution

```r
#nhanes
imps <- mice(nhanes,method="norm",printFlag=FALSE,seed=1)
fits <- with(imps,lm(bmi~age+hyp+chl))
ests <- pool(fits)
print(ests[,3][,c(3,4,5,9,10)])
```

```
##      estimate         ubar            b       riv    lambda
## 1 18.62571842 8.0320619776 1.9252212801 0.2876304 0.2233797
## 2 -6.15589205 0.9691053340 1.2101679813 1.4984971 0.5997594
## 3  1.27329758 2.1309028741 1.6772372744 0.9445220 0.4857348
## 4  0.08878308 0.0002998066 0.0001824689 0.7303465 0.4220811
```

```r
#summary(ests,conf.int=TRUE)[,c(2,3,6,7,8)]
```

From the pooled estimates, the proportions of variance due to missing data($\frac{B+\frac{B}{M}}{V^{\top}}$) for intercept is 0.2233797, for the coefficient of “age” is 0.5997594, for the coefficient of “hyp” is 0.4857348, for the coefficient of “chl” is 0.4220811. Looking at the riv, *relative increase in variance*($\frac{B+\frac{B}{M}}{U}$). “riv” of the coefficient of “age” is the largest. Therefore, the parameter for “age” appear to be most affected by the nonresponse.

(*c*) My solution

```
ests_2 <- pool(with(mice(nhanes,method="norm",printFlag=FALSE,seed=2),lm(bmi~age+hyp+chl)))
ests_3 <- pool(with(mice(nhanes,method="norm",printFlag=FALSE,seed=3),lm(bmi~age+hyp+chl)))
ests_4 <- pool(with(mice(nhanes,method="norm",printFlag=FALSE,seed=4),lm(bmi~age+hyp+chl)))
ests_5 <- pool(with(mice(nhanes,method="norm",printFlag=FALSE,seed=5),lm(bmi~age+hyp+chl)))
ests_6 <- pool(with(mice(nhanes,method="norm",printFlag=FALSE,seed=6),lm(bmi~age+hyp+chl)))
ests_2[,3][,c(3,4,5,9,10)]
```

```
##      estimate         ubar            b      riv    lambda
## 1 18.77930741 6.3548310733 5.3038256200 1.001536 0.5003837
## 2 -4.91871335 0.7145833610 1.9260762104 3.234460 0.7638424
## 3  1.58775477 2.2527905073 2.7801835767 1.480928 0.5969250
## 4  0.07288751 0.0002110306 0.0002685269 1.526946 0.6042654
```

```
ests_3[,3][,c(3,4,5,9,10)]
```

```
##      estimate         ubar            b       riv    lambda
## 1 16.64286080 9.7691595817 1.9980081281 0.2454264 0.1970622
## 2 -6.07984016 1.0658102179 0.2410877311 0.2714416 0.2134912
## 3  2.42087526 3.0357823167 3.0250718110 1.1957663 0.5445781
## 4  0.09260882 0.0002701174 0.0002072138 0.9205499 0.4793158
```

```
ests_4[,3][,c(3,4,5,9,10)]
```

```
##      estimate         ubar           b       riv    lambda
## 1 19.12182720 7.1765222903 5.830375722 0.9749083 0.4936474
## 2 -5.30149475 1.0119962484 0.562070207 0.6664889 0.3999360
## 3  1.10872245 2.8720799677 2.696892338 1.1268039 0.5298109
## 4  0.07779401 0.0002621676 0.000127816 0.5850428 0.3691022
```

```
ests_5[,3][,c(3,4,5,9,10)]
```

```
##      estimate         ubar            b       riv    lambda
## 1 19.12666914 7.6630181535 0.7356811697 0.1152049 0.1033038
## 2 -5.30691629 1.0036805866 0.5108135923 0.6107285 0.3791629
## 3  1.85855685 2.9852865611 0.8835272900 0.3551528 0.2620758
## 4  0.07397953 0.0002259983 0.0001586056 0.8421600 0.4571590
```

```
ests_6[,3][,c(3,4,5,9,10)]
```

```
##      estimate         ubar            b       riv    lambda
## 1 21.92213191 7.0749830341 1.057230e+01 1.7931857 0.6419858
## 2 -4.75839950 1.0631296436 1.003823e+00 1.1330585 0.5311896
## 3  0.61823443 3.2432094556 2.672881e+00 0.9889763 0.4972288
## 4  0.06185465 0.0002590881 3.420285e-04 1.5841496 0.6130255
```

(*d*) **My solution**

```r
ests_m1 <- pool(with(mice(nhanes,method="norm",printFlag=FALSE,seed=1,m=100),lm(bmi~age+hyp+chl)))
ests_m2 <- pool(with(mice(nhanes,method="norm",printFlag=FALSE,seed=2,m=100),lm(bmi~age+hyp+chl)))
ests_m3 <- pool(with(mice(nhanes,method="norm",printFlag=FALSE,seed=3,m=100),lm(bmi~age+hyp+chl)))
ests_m4 <- pool(with(mice(nhanes,method="norm",printFlag=FALSE,seed=4,m=100),lm(bmi~age+hyp+chl)))
ests_m5 <- pool(with(mice(nhanes,method="norm",printFlag=FALSE,seed=5,m=100),lm(bmi~age+hyp+chl)))
ests_m6 <- pool(with(mice(nhanes,method="norm",printFlag=FALSE,seed=6,m=100),lm(bmi~age+hyp+chl)))
ests_m1[,3][,c(3,4,5,10)]
```

```
##      estimate         ubar            b    lambda
## 1 19.50578464 7.6726982210 4.8232234828 0.3883447
## 2 -5.01414576 0.9377506602 1.1370603692 0.5504942
## 3  1.79653382 2.3071645301 2.4969648822 0.5222371
## 4  0.07000477 0.0002442783 0.0002188093 0.4749816
```

```r
ests_m2[,3][,c(3,4,5,10)]
```

```
##      estimate        ubar            b    lambda
## 1 19.30211794 8.160301562 5.0531412996 0.3847770
## 2 -5.22461768 0.994794777 1.2413355845 0.5575826
## 3  1.80604694 2.585999019 2.8077380292 0.5230381
## 4  0.07288669 0.000279477 0.0002539153 0.4785210
```

```r
ests_m3[,3][,c(3,4,5,10)]
```

```
##      estimate         ubar            b    lambda
## 1 19.35383641 8.0416025218 8.0626341605 0.5031405
## 2 -5.18649404 0.9996955427 1.1662373830 0.5409177
## 3  2.01676543 2.7618714323 2.9813449886 0.5215907
## 4  0.07059777 0.0002690501 0.0002878005 0.5193204
```

```r
ests_m4[,3][,c(3,4,5,10)]
```

```
##      estimate         ubar            b    lambda
## 1 19.82556330 7.7408772244 7.4062178664 0.4914396
## 2 -5.15892718 0.9471221433 1.1018091293 0.5402207
## 3  1.90013019 2.4905489867 2.8481881225 0.5359703
## 4  0.06900731 0.0002485457 0.0002908957 0.5417248
```

```r
ests_m5[,3][,c(3,4,5,10)]
```

```
##      estimate         ubar            b    lambda
## 1 19.28450230 7.7442779668 5.7490522338 0.4285012
## 2 -5.24878515 0.9537854913 1.1340454687 0.5456372
## 3  1.56462784 2.5694549372 2.2863326879 0.4733268
## 4  0.07462768 0.0002583617 0.0002954493 0.5359596
```

```r
ests_m6[,3][,c(3,4,5,10)]
```

```
##     estimate         ubar           b    lambda
## 1 19.2738085 8.8310791310 5.635913639 0.3919393
## 2 -5.2791050 1.0766589491 1.114595662 0.5111430
## 3  1.9327925 2.7973782943 3.209178305 0.5367542
## 4  0.0726744 0.0002773879 0.000276356 0.5015559
```

Question 2

My solution

```
load("dataex2.Rdata")

# built a function to return a matrix with a row 95% confidence intervals for beta_1
# input:  ## D, 2-D, from dataex2
          ## Mtd, a string used in mice() function
# output: ## a matrix
CI_beta1 <- function(D,Mtd){
  imps <- mice(data=D,method=Mtd,printFlag=FALSE,seed=1,m=20)
  ests <- pool(with(imps,lm(Y~X)))
  return(matrix(summary(ests,conf.int=TRUE)[,c(7,8)][2,],nrow=1))
}

# built a function to return a matrix with rows 95% confidence intervals for beta_1
# input:  ## data, 3-D,initialized as dataex2
          ## Mtd, a string used in mice() function
# output: ## a matrix
CIs <- function(data=dataex2,Mtd){
  Conf <- matrix(nrow=1,ncol=2)
  for (i in 1:100){
    A <- CI_beta1(data[,,i],Mtd=Mtd)
    Conf <- rbind(Conf,A)
  }
  return(Conf[2:101,])
}

# function to return a emprical coverage probability
  ## with 95% confidence intervals for beta_1
# input:  ## data, 3-D,initialized as dataex2
          ## Mtd, a string used in mice() function
# output: ## emprical coverage probability(numeric)
ECP <- function(data=dataex2,Mtd){
  n = 100; times = 0
  B = CIs(Mtd=Mtd)
  for (i in 1:100){
    times = ifelse(B[i,][1]<3 & B[i,][2]>3,times+1,times)
  }
  return(times/n)
}

print(c("The emprical probability for ß1 using stochastic regression imputation is",
        ECP(Mtd="norm.nob")),quote=FALSE)
```

```
## [1] The emprical probability for ß1 using stochastic regression imputation is
## [2] 0.88
```

```
print(c("The emprical probability for ß1 using the corresponding bootstrap based version is",
        ECP(Mtd="norm.boot")),quote=FALSE)
```

```
## [1] The emprical probability for ß1 using the corresponding bootstrap based version is
## [2] 0.95
```

Question 3

My solution

Question 4

(a) My solution

```
load("dataex4.Rdata")
imps <- mice(dataex4,printFlag=FALSE,seed=1,m=50)
ests <- pool(with(imps,lm(y~x1*x2)))
summary(ests,conf.int=TRUE)[,c(1,2,7,8)]
```

```
##          term  estimate    2.5 %    97.5 %
## 1 (Intercept) 1.5929831 1.404501 1.7814655
## 2          x1 1.4112333 1.219397 1.6030697
## 3          x2 1.9658191 1.860657 2.0709812
## 4       x1:x2 0.7550367 0.642302 0.8677715
```

The estimates for β_1 is 1.4112333, and the 95% confident interval is $[1.219397, 1.6030697]$;

The estimates for β_1 is 1.9658191, and the 95% confident interval is $[1.860657, 2.0709812]$;

The estimates for β_1 is 0.7550367, and the 95% confident interval is $[0.642302, 0.8677715]$.

(b) My solution

```
data4 =
  dataex4 %>%
  mutate(inter = x1*x2)

#data4
imps <- mice(data4,printFlag=FALSE,seed=1,m=50)
# change the method using I() method
meth <- imps$method
meth["inter"] <- "~I(x1*x2)"

# prevent feedback from interaction in the imputation of x1 and x2
pred <- imps$predictorMatrix
# x1*x2 will not be used as predictor of x1 and x2
pred[c("x1", "x2"), "inter"] <- 0
pred[,c("x1","x2")] <- 0
pred["x1","x2"] <- 1
pred["x2","x1"] <- 1

# make sure x1*x2 ordered at last
visSeq <- imps$visitSequence
```

```r
which_inter <- match("inter", visSeq)
visSeq <- c(visSeq[-which_inter], visSeq[which_inter])

# passive imputation to impute the interaction variable
imp <- mice(data4,method=meth,predictorMatrix=pred,visitSequence=visSeq,m=50,seed=1,printFlag = FALSE)
ests <- pool(with(imp,lm(y~x1*x2)))
summary(ests,conf.int=TRUE)[,c(1,2,7,8)]
```

```
##          term  estimate     2.5 %   97.5 %
## 1 (Intercept) 2.1654541 1.8968644 2.434044
## 2          x1 0.9761881 0.6992222 1.253154
## 3          x2 1.6168272 1.4688180 1.764836
## 4       x1:x2 0.9470357 0.7999456 1.094126
```

```r
## check problems mice() detected
imp$loggedEvents
```

```
## NULL
```

The estimates for β_1 is 0.9761881, and the 95% confident interval is $[0.6992222, 1.253154]$;

The estimates for β_1 is 1.6168272, and the 95% confident interval is $[1.4688180, 1.764836]$;

The estimates for β_1 is 0.9470357, and the 95% confident interval is $[0.7999456, 1.094126]$.

(c) **My solution**

```r
imp <- mice(data4,method=meth,m=50,seed=1,printFlag = FALSE)
ests <- pool(with(imp,lm(y~x1+x2+inter)))
summary(ests,conf.int=TRUE)[,c(1,2,7,8)]
```

```
##          term  estimate     2.5 %    97.5 %
## 1 (Intercept) 1.5722935 1.4036067 1.7409803
## 2          x1 1.2657606 1.0714517 1.4600696
## 3          x2 1.9826229 1.8858124 2.0794333
## 4       inter 0.8022453 0.6865434 0.9179472
```

The estimates for β_1 is 1.2657606, and the 95% confident interval is $[1.0714517, 1.4600696]$;

The estimates for β_1 is 1.9826229, and the 95% confident interval is $[1.8858124, 2.0794333]$;

The estimates for β_1 is 0.8022453, and the 95% confident interval is $[0.6865434, 0.9179472]$.

(d) **My solution**

```r
imp$predictorMatrix
```

```
##       y x1 x2 inter
## y     0  1  1     1
## x1    1  0  1     1
## x2    1  1  0     1
## inter 1  1  1     0
```

It could cause multi-colinearity.

Question 5

My solution

Step1: Inspect To be started, using the dim() method we see that there are 500 rows, and 12 variables.

```
load('NHANES2.Rdata')
dim(NHANES2)
```

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

further inspect the nature of our variables and check they are correctly coded.

```
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 ...
```

Check the simple statistics (min/max/mean/quantiles) of the observed data in each variable along with the number of missing values by summary() method. As we can see, “wgt”, “age”, and “race” are complete data.

```
summary(NHANES2)
```

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

Then inspect the missing pattern of the data.

```
require(JointAI)
```

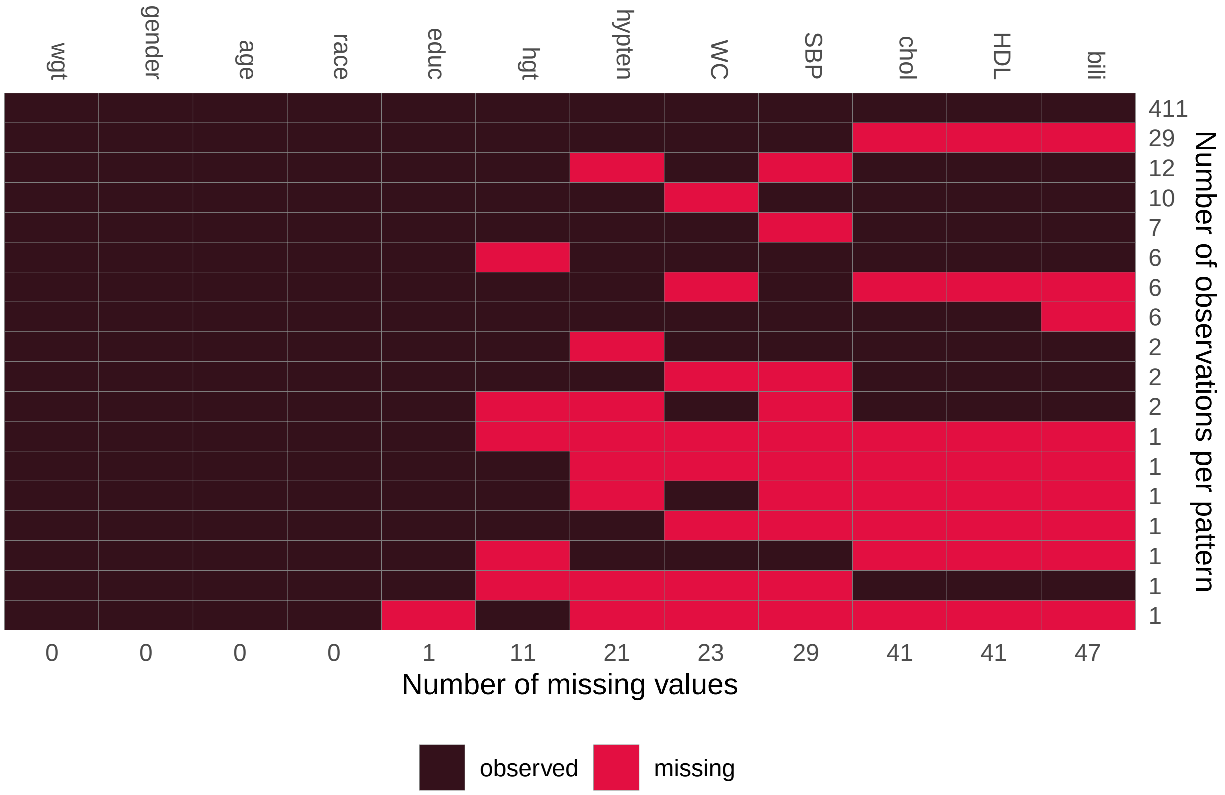
```
## Loading required package: JointAI
```

```
## Please report any bugs to the package maintainer (https://github.com/NErler/JointAI/issues).
```

```
##
## Attaching package: 'JointAI'
```

```
## The following object is masked from 'package:dplyr':
##
##     all_vars
```

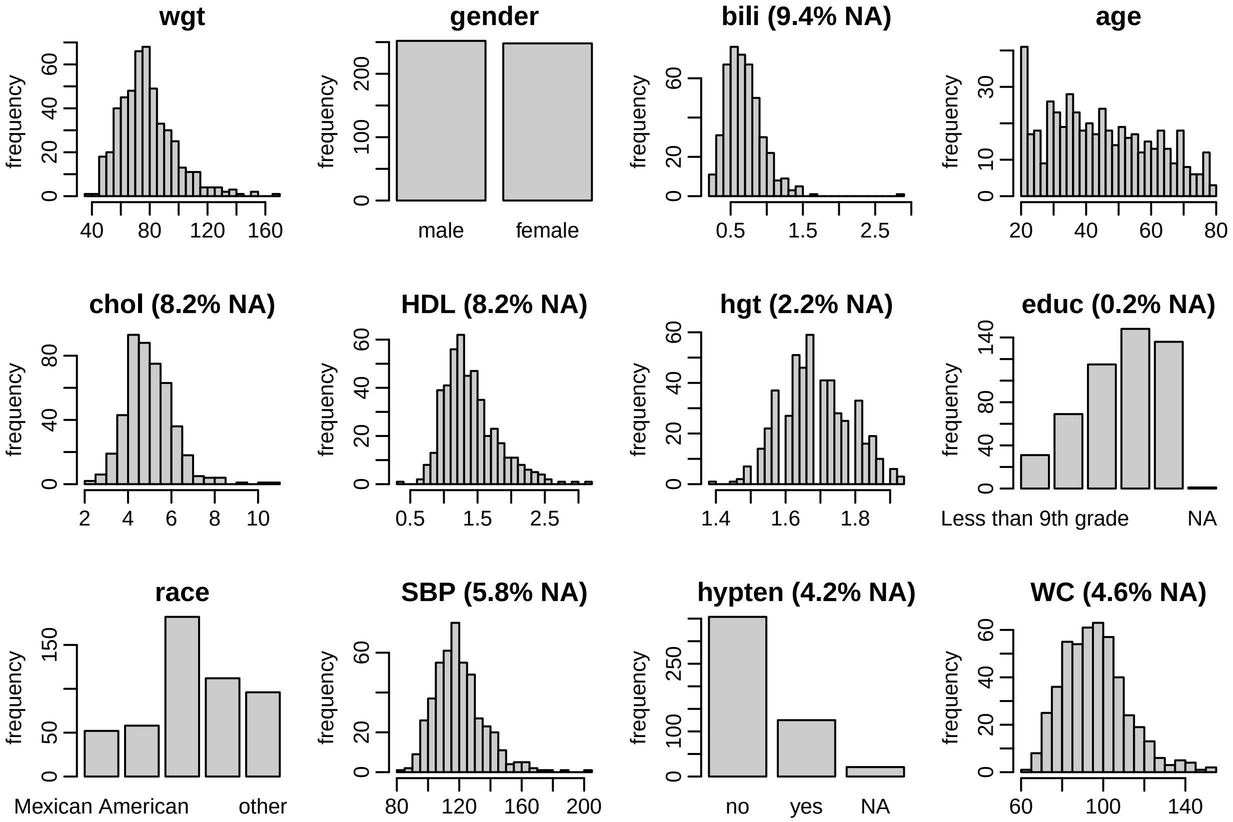
```
md_pattern(NHANES2, pattern = FALSE, color = c('#34111b', '#e30f41'))
```



Learnt from the chart showed above, there are 411 observations with observed values on all 12 variables. Also, 29 observations for which bilirubin concentration in mg/dL, High-density lipoprotein cholesterol in mg/dL, and otal serum cholesterol in mg/dL are missing, etc.

Visualise the obeserved data in the missing dataset by pacakage JointAI to see if there is normality between varaibles.

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



##### Step2: Imputation