Assignment 3

Zekai Li S2040608

Question 1

(*a*) My solution

```
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

```
#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
```

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
```

From the summary of the same model using different seeds, the results remaint the same.

(*d*) **My solution**

```
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)))
summary(ests_m1,conf.int=TRUE)
```

```
##          term    estimate  std.error  statistic        df      p.value
## 1 (Intercept) 19.50578464 3.54177271  5.5073508 11.566895 0.0001537309
## 2         age -5.01414576 1.44436202 -3.4715298  8.429706 0.0077703513
## 3         hyp  1.79653382 2.19752112  0.8175274  8.969678 0.4348045440
## 4         chl  0.07000477 0.02157025  3.2454315  9.879073 0.0089256326
##         2.5 %     97.5 %
## 1 11.75675587 27.2548134
## 2 -8.31552758 -1.7127639
## 3 -3.17716718  6.7702348
## 4  0.02186339  0.1181461
```

```
summary(ests_m2,conf.int=TRUE)
```

```
##          term    estimate  std.error  statistic        df      p.value
## 1 (Intercept) 19.30211794 3.64197395  5.2999055 11.636939 0.0002088951
## 2         age -5.22461768 1.49951449 -3.4842062  8.294691 0.0078155906
## 3         hyp  1.80604694 2.32847899  0.7756338  8.954332 0.4579599106
## 4         chl  0.07288669 0.02315019  3.1484270  9.810681 0.0105993881
##         2.5 %     97.5 %
## 1 11.33939340 27.2648425
## 2 -8.66123676 -1.7879986
## 3 -3.46543638  7.0775303
## 4  0.02116961  0.1246038
```

```
summary(ests_m3,conf.int=TRUE)
```

```
##          term    estimate  std.error  statistic       df      p.value
## 1 (Intercept) 19.35383641 4.02304151  4.8107474 9.336207 0.0008647648
## 2         age -5.18649404 1.47566775 -3.5146760 8.612392 0.0070378035
## 3         hyp  2.01676543 2.40271302  0.8393701 8.982063 0.4230385677
## 4         chl  0.07059777 0.02365858  2.9840237 9.025573 0.0153016849
##         2.5 %     97.5 %
## 1 10.30279748 28.4048753
## 2 -8.54772901 -1.8252591
## 3 -3.42020403  7.4537349
## 4  0.01710145  0.1240941
```

```
summary(ests_m4,conf.int=TRUE)
```

```
##          term    estimate  std.error  statistic       df      p.value
## 1 (Intercept) 19.82556330 3.90143016  5.0816143 9.561437 0.0005480812
## 2         age -5.15892718 1.43525237 -3.5944391 8.625701 0.0062155718
## 3         hyp  1.90013019 2.31672592  0.8201791 8.706895 0.4339921276
## 4         chl  0.06900731 0.02328842  2.9631602 8.596982 0.0166834062
##         2.5 %     97.5 %
## 1 11.07828404 28.5728426
## 2 -8.42729747 -1.8905569
## 3 -3.36768488  7.1679453
## 4  0.01594637  0.1220683
```

```r
summary(ests_m5,conf.int=TRUE)
```

```
##          term    estimate  std.error  statistic        df      p.value
## 1 (Intercept) 19.28450230 3.68114394  5.2387254 10.781369 0.0002963267
## 2         age -5.24878515 1.44885176 -3.6227206  8.522322 0.0060750438
## 3         hyp  1.56462784 2.20876684  0.7083717  9.911066 0.4950372482
## 4         chl  0.07462768 0.02359588  3.1627420  8.707099 0.0119833676
##         2.5 %     97.5 %
## 1 11.16226835 27.4067363
## 2 -8.55453384 -1.9430365
## 3 -3.36280392  6.4920596
## 4  0.02097512  0.1282802
```

```r
summary(ests_m6,conf.int=TRUE)
```

```
##          term   estimate  std.error  statistic        df      p.value
## 1 (Intercept) 19.2738085 3.81095157  5.0574792 11.496363 0.0003208074
## 2         age -5.2791050 1.48404871 -3.5572316  9.182452 0.0059527100
## 3         hyp  1.9327925 2.45736615  0.7865301  8.691915 0.4524665449
## 4         chl  0.0726744 0.02359041  3.0806754  9.366681 0.0125301040
##         2.5 %     97.5 %
## 1 10.92993437 27.6176826
## 2 -8.62611561 -1.9320944
## 3 -3.65633685  7.5219218
## 4  0.01962595  0.1257229
```

The (pooled) estimates, standard errors, and the bounds of the intervals get more stable as M increases and we can be more confident in any one specific run.

Question 2

My solution

```r
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
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
```

```
## 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**

```
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**

```
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
```

From the Predictor Matrix, we know that it is not modified, this will accompany with a high risk of 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.

```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 ...
```

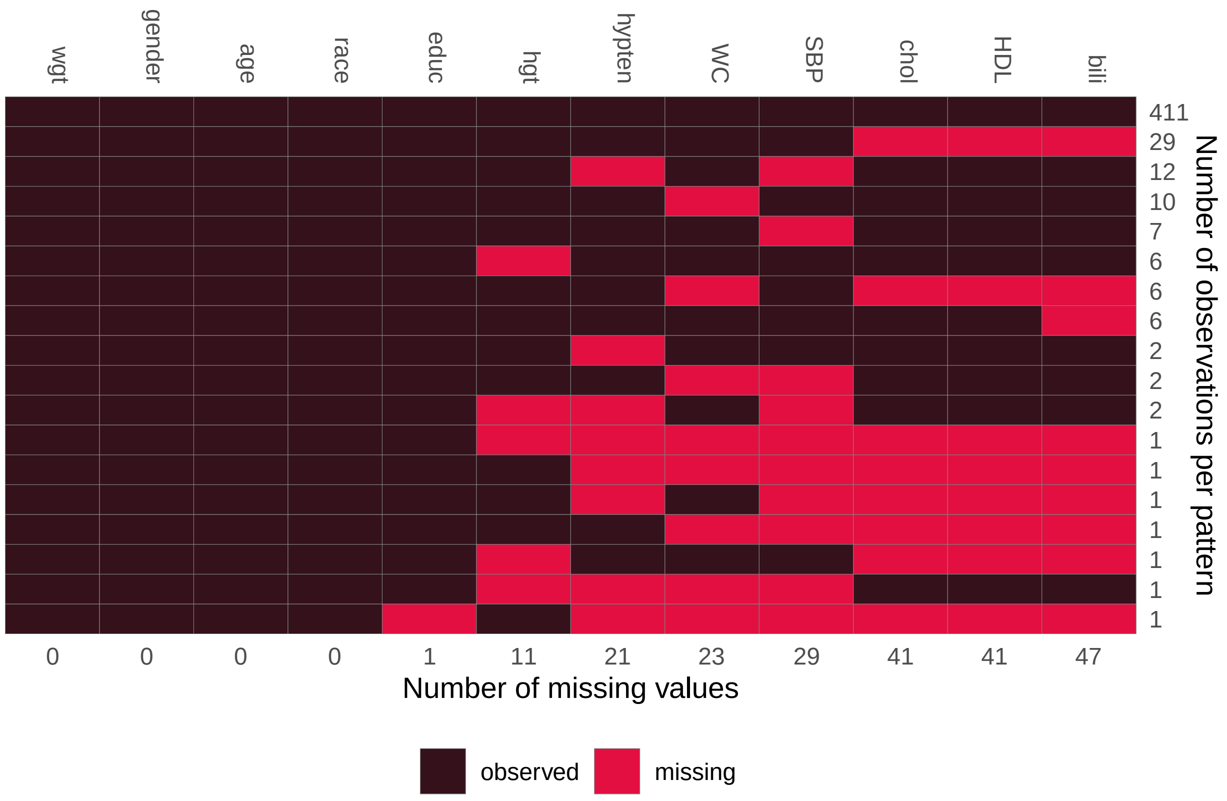
Check the simple statistics (min/max/mean/quantiles) of the observed data in each variable along with the number of missing values by summary() method. As we can see, “wgt”, “age”, and “race” are complete data.

```r
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.(using pacakage JointAI)

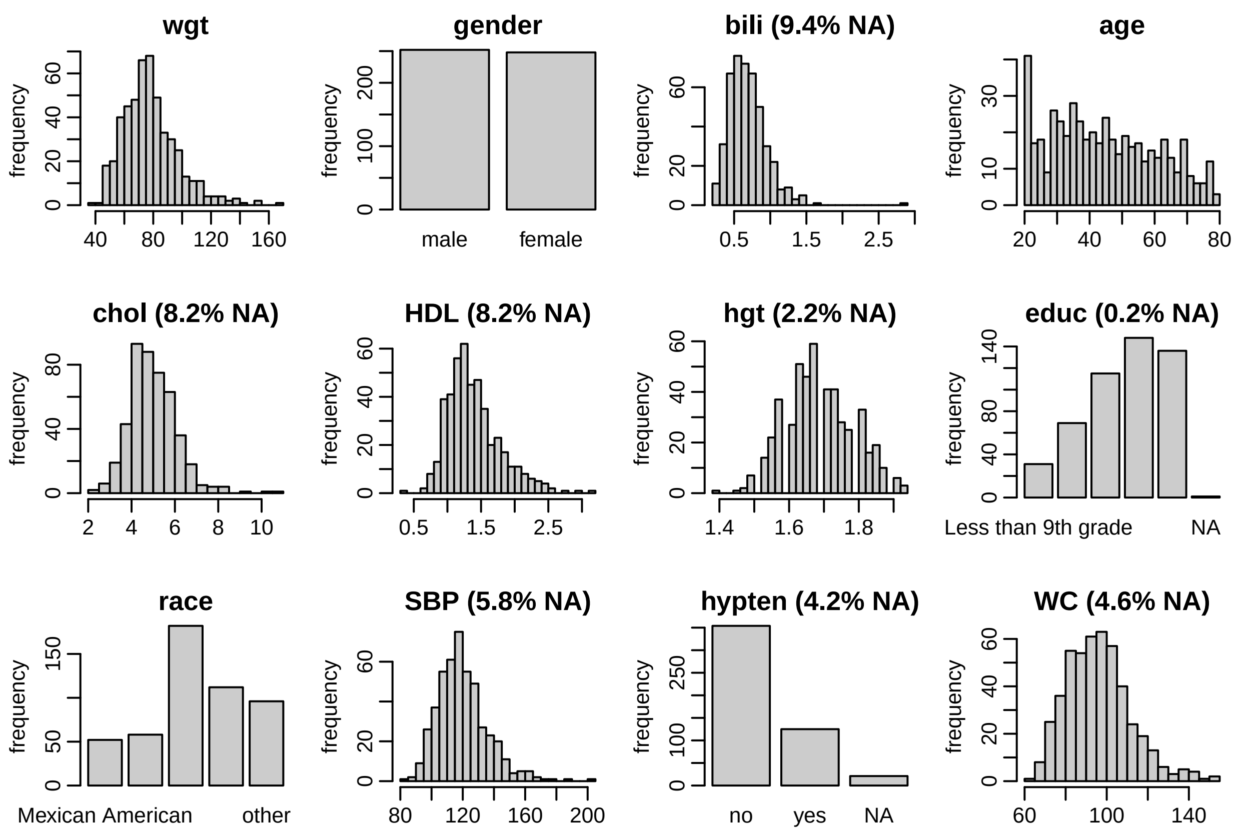
```r
md_pattern(NHANES2, pattern = FALSE, color = c('#34111b', '#e30f41'))
```



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)
```



Step1: Imputation Using the default seeting in the mice() method to see what will happen.

```r
imp0 <- mice(NHANES2, maxit = 0)
imp0
```

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

From the distribution plot depicting in previous step, It is not unreasonable to change the default imputation method from pmm to norm for the variable “bili”, “chol”, “HDL”, “hgt”, “SBP”, and “WC”.

```
meth <- imp0$method
meth[c("bili", "chol", "HDL", "hgt", "SBP", "WC")] <- "norm"
meth
```

```
##      wgt   gender     bili      age     chol      HDL      hgt     educ
##       ""       ""   "norm"       ""   "norm"   "norm"   "norm"   "polr"
##     race      SBP   hypten       WC
##       ""   "norm" "logreg"   "norm"
```

For these varibles("bili", "chol", "HDL", "hgt", "SBP", and "WC"), there are risks to impute negative values.

```
post <- imp0$post
post["bili"] <- "imp[[j]][,i] <- squeeze(imp[[j]][,i], c(0, 20))"
post["chol"] <- "imp[[j]][,i] <- squeeze(imp[[j]][,i], c(0, 100))"
post["HDL"] <- "imp[[j]][,i] <- squeeze(imp[[j]][,i], c(0, 100))"
post["hgt"] <- "imp[[j]][,i] <- squeeze(imp[[j]][,i], c(0, 3))"
post["SBP"] <- "imp[[j]][,i] <- squeeze(imp[[j]][,i], c(0, 300))"
post["WC"] <- "imp[[j]][,i] <- squeeze(imp[[j]][,i], c(0, 200))"
```

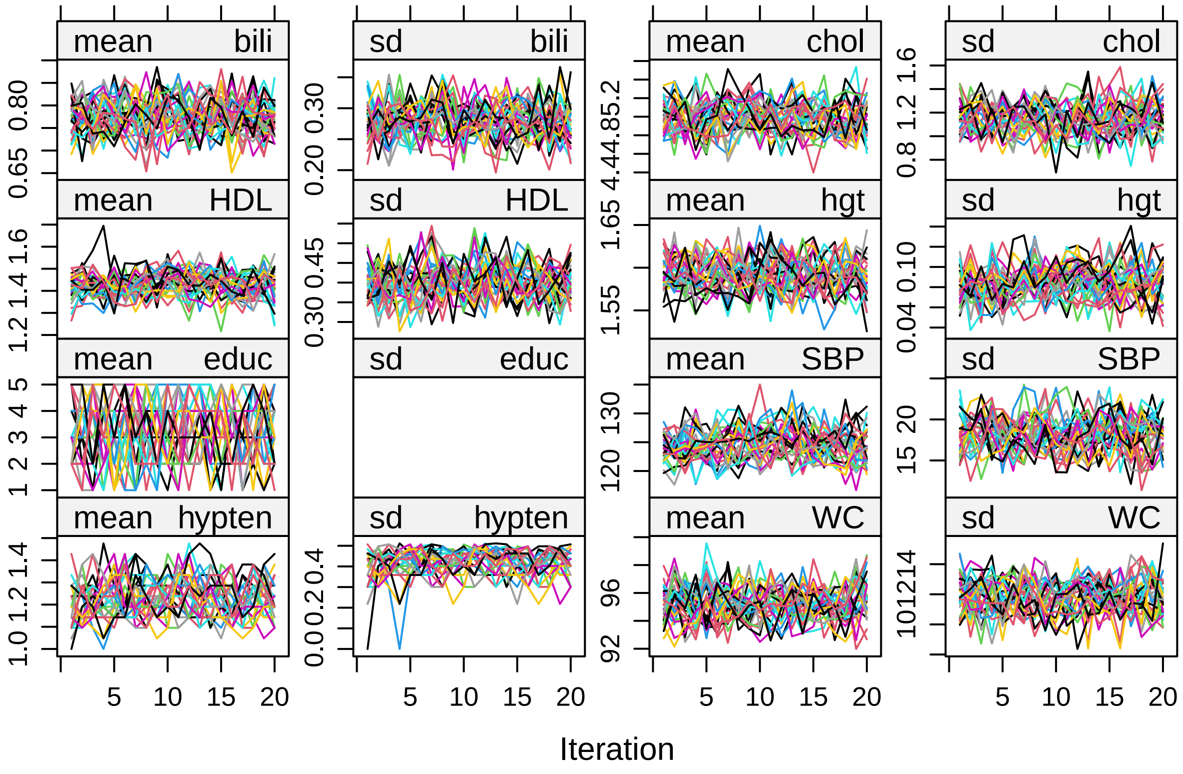
Then go for imputation:

```
imp <- mice(NHANES2,method = meth,post=post,maxit=20,m=30,seed=1,printFlag=FALSE)
## check problems mice() detected
imp$loggedEvents
```

```
## NULL
```

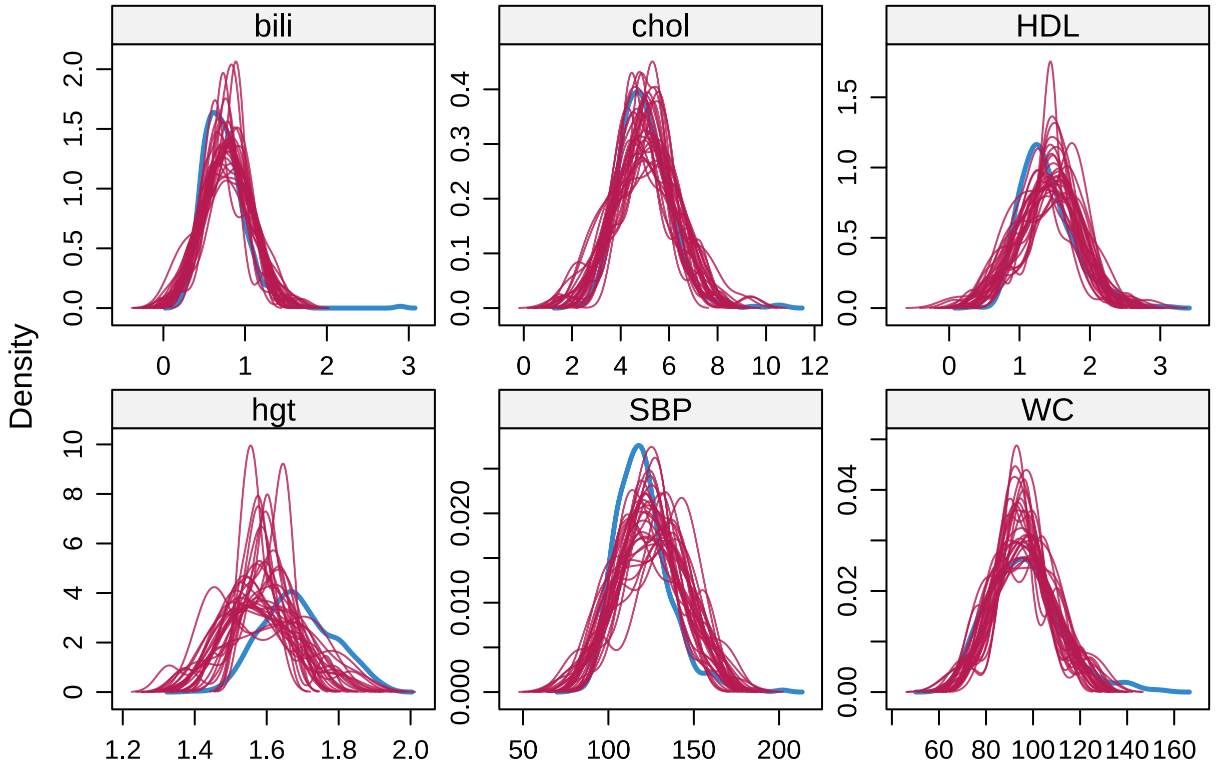
Before fitting the model, we have to check the convergence. First we need to visulise the traceplot.

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



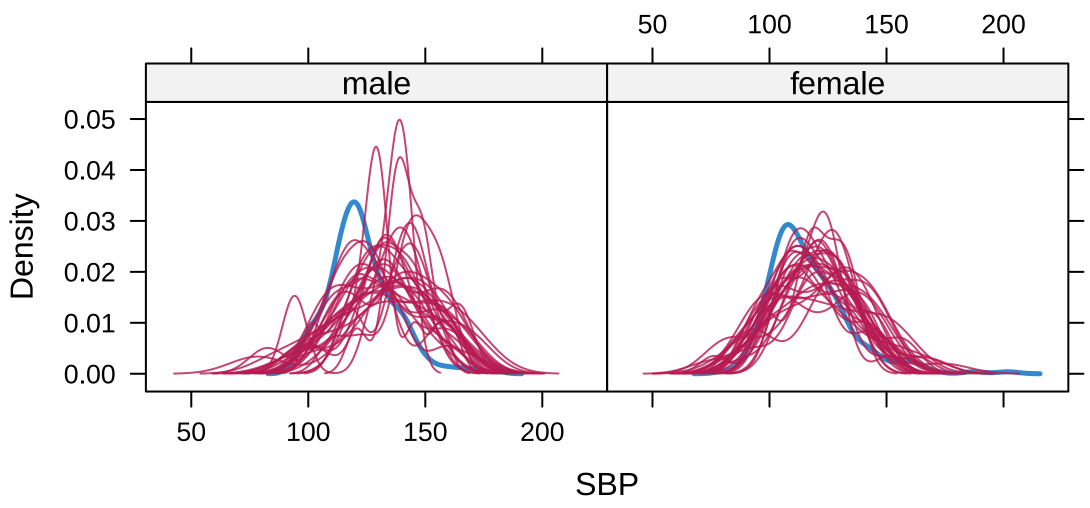
from the chart above, we learnt that the iterative algorithm appears to have converged for all variables that were imputed. Then we compare the distribution of the imputed values against the distribution of the observed values. We start doing that for the continuous variables.

```
densityplot(imp)
```

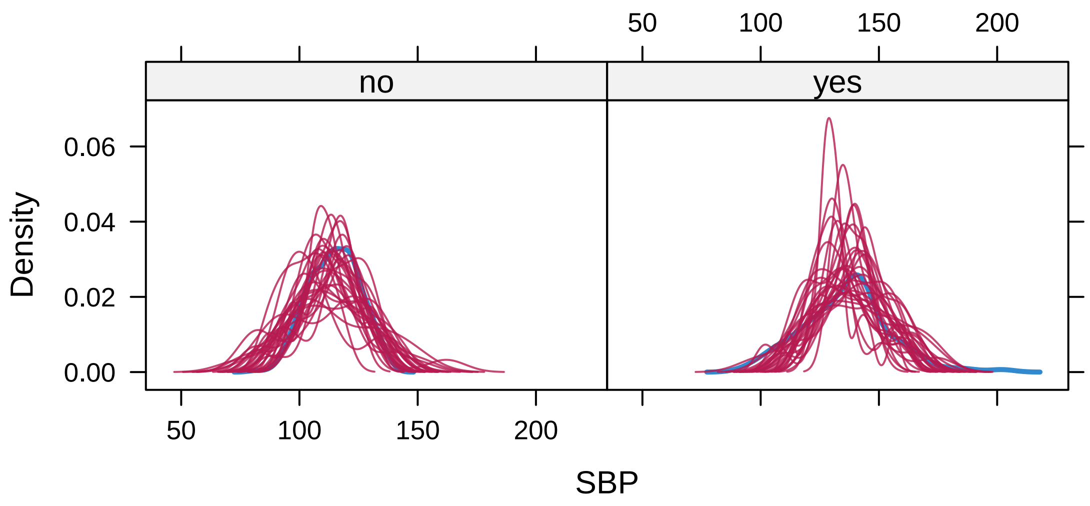


Above, there are 30 red lines which are imputed values, and the the blue ones are from observed data. plots of "chol" and "SBP" perform best. Then check SBP conditional on the gender, and hypertensive status and height conditional on gender.

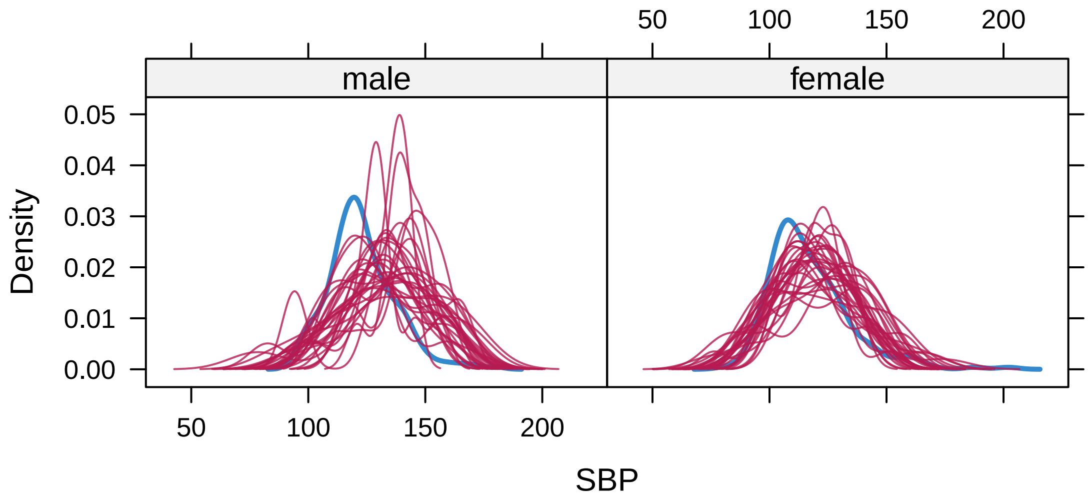
```r
densityplot(imp, ~SBP|gender)
```



```r
densityplot(imp, ~SBP|hypten)
```



```r
densityplot(imp, ~SBP|gender)
```

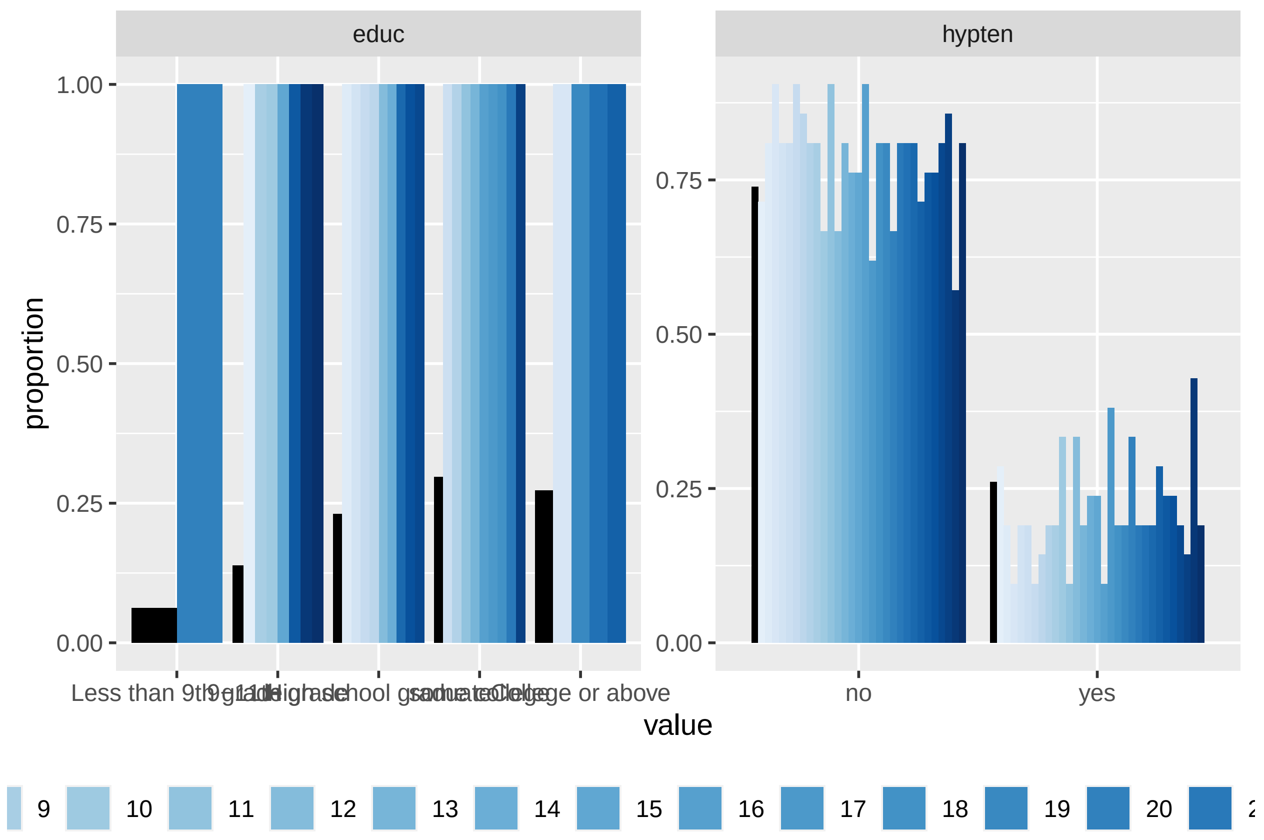


From the conditional charts, it seems that differences between the observed and imputed values for SBP affected by gender and hypertensive status, while it seems hard to be seen the relation for the variable height.

```
source_url("https://gist.githubusercontent.com/NErler/0d00375da460dd33839b98faeee2fdab/raw/c6f537ecf80e
```

```
FALSE SHA-1 hash of file is f8e69c429689d084809da7ee99db4411def17c68
```

```
propplot(imp)
```



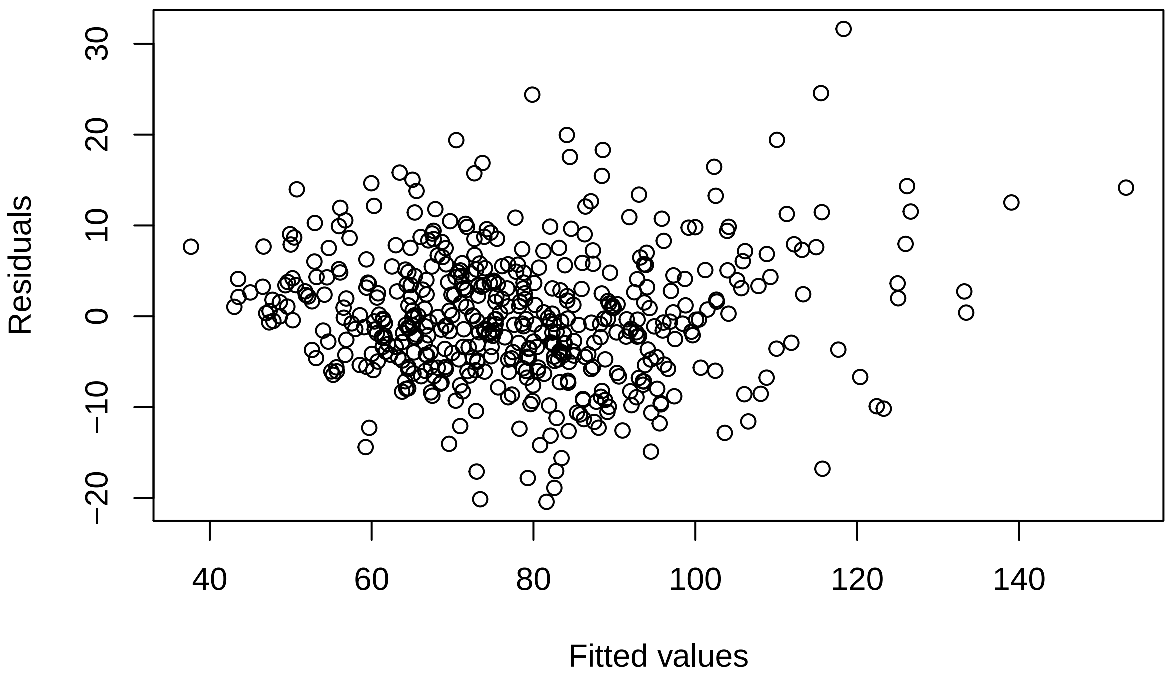
Observed the chart above, there is a large discrepancy between imputed and obeserved data for varible “educ”(educational status), but we do not worry about it for “educ” only has one missing value.

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

Step 3: Fitting values

```
##
## Call:
## lm(formula = wgt ~ gender + age + hgt + WC)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.417  -4.575  -0.460   4.087  31.629
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -100.33285    7.41999 -13.522  < 2e-16 ***
## genderfemale    -1.26941    0.82473  -1.539    0.124
## age             -0.16023    0.02092  -7.660 9.89e-14 ***
## hgt             52.08561    4.23931  12.286  < 2e-16 ***
## WC               1.02760    0.02219  46.315  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.194 on 495 degrees of freedom
## Multiple R-squared:  0.8569, Adjusted R-squared:  0.8558
## F-statistic: 741.1 on 4 and 495 DF,  p-value: < 2.2e-16
```

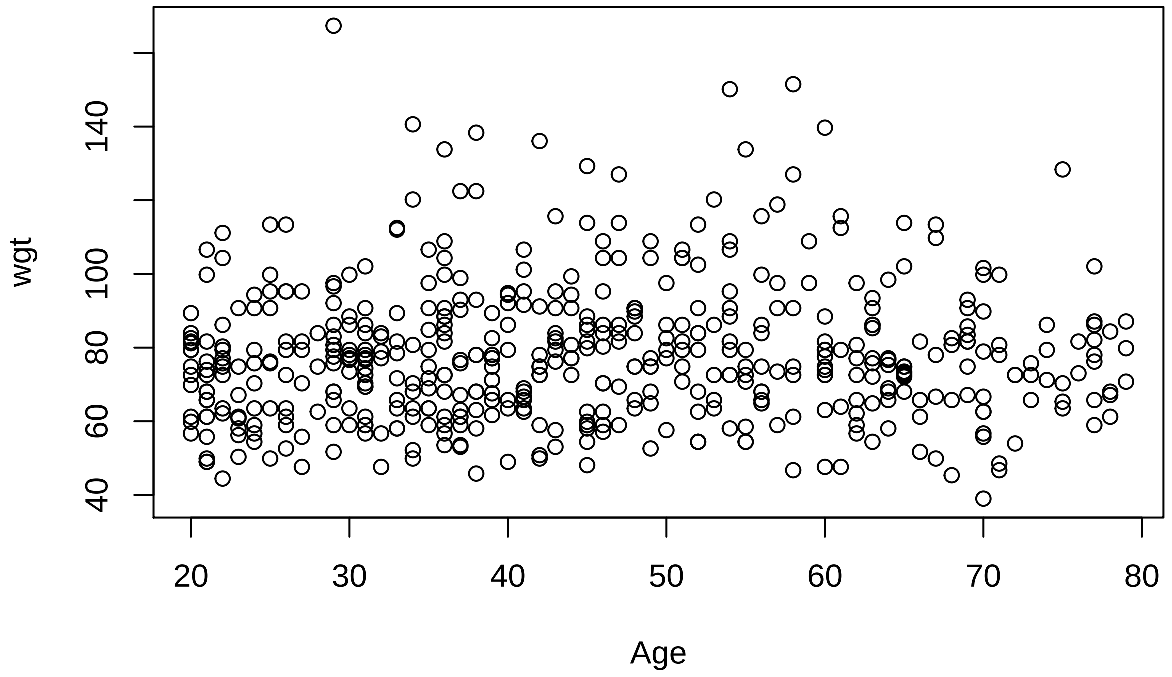
```r
comp1 <- complete(imp, 1)
ind <- sample(1:30,1)
plot(fit$analyses[[ind]]$fitted.values, residuals(fit$analyses[[1]]),
     xlab = "Fitted values", ylab = "Residuals")
```



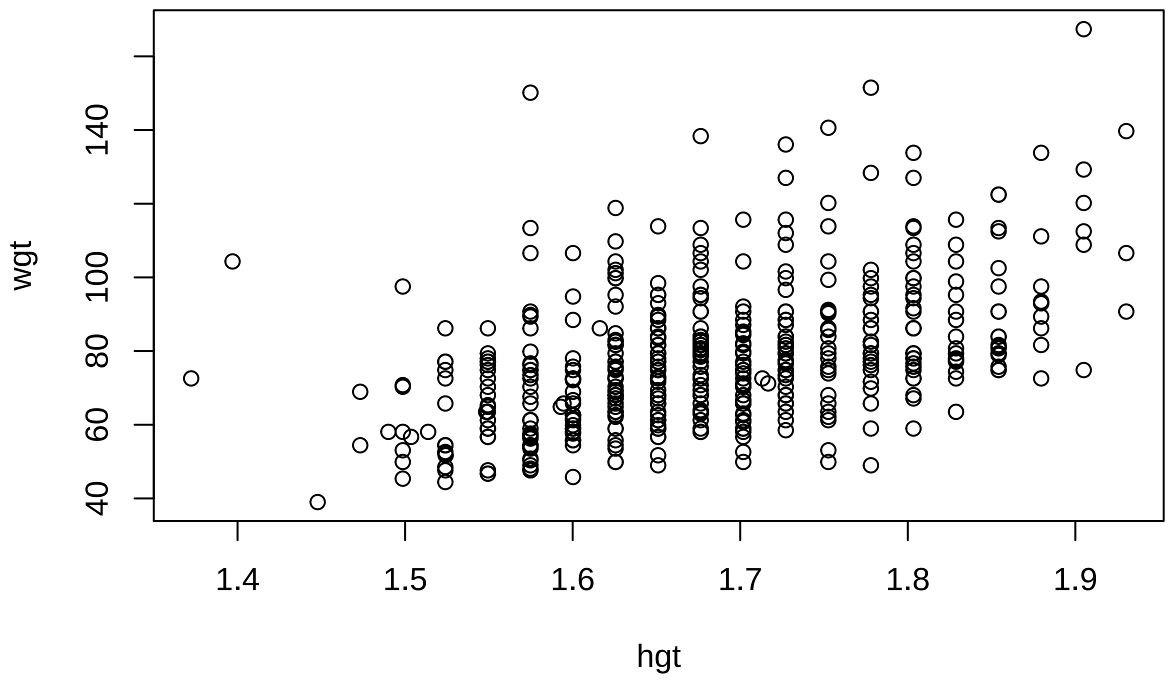
The residual plot shows a little messy, but it cannot be enough to say it is against the homoscedastic assumption.

Seperately compare the response with other variables, we can find that weight do not show some tendency with the increase of age, but there are obvious positive relation with height and waist circumference, which are intuitive. And the box plot illustrates that male heavier than female on average.

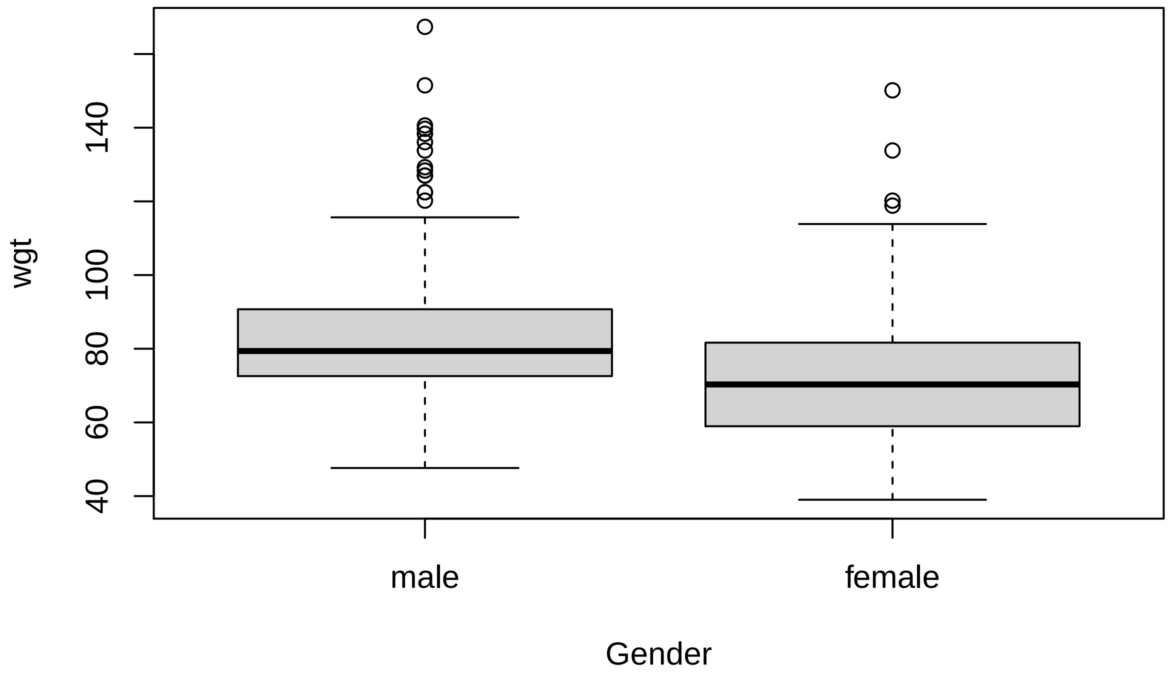
```
plot(comp1$wgt ~ comp1$age, xlab = "Age", ylab = "wgt")
```



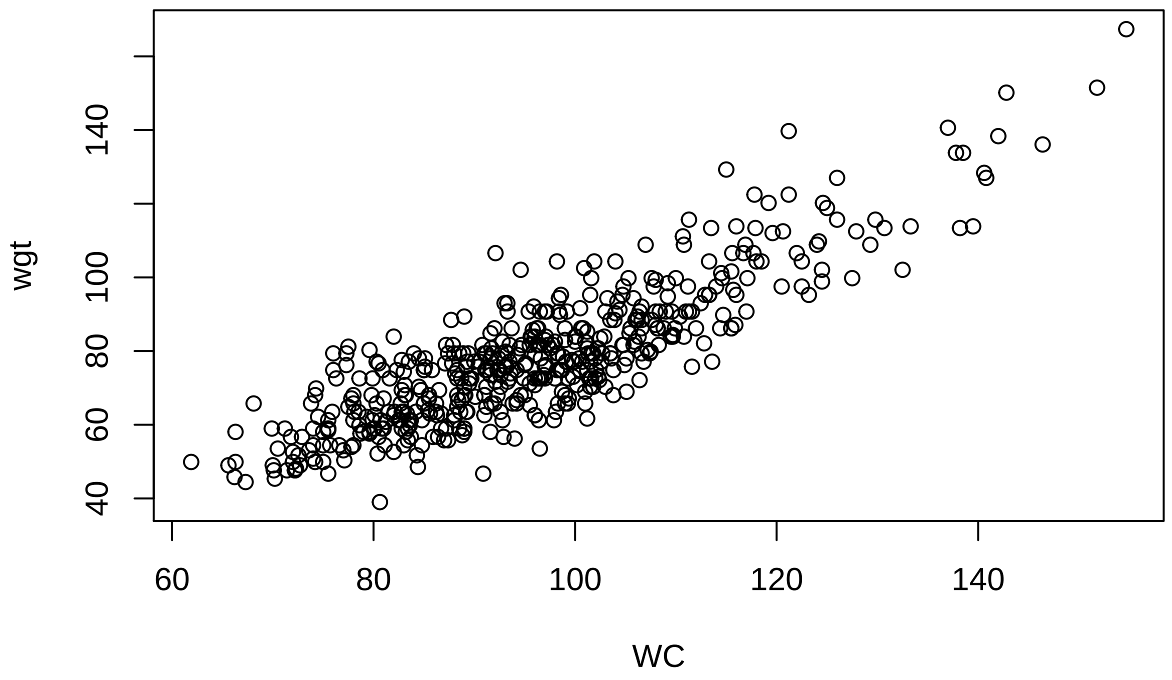
```
plot(comp1$wgt ~ comp1$hgt, xlab = "hgt", ylab = "wgt")
```



```r
boxplot(comp1$wgt ~ comp1$gender, xlab = "Gender", ylab = "wgt")
```

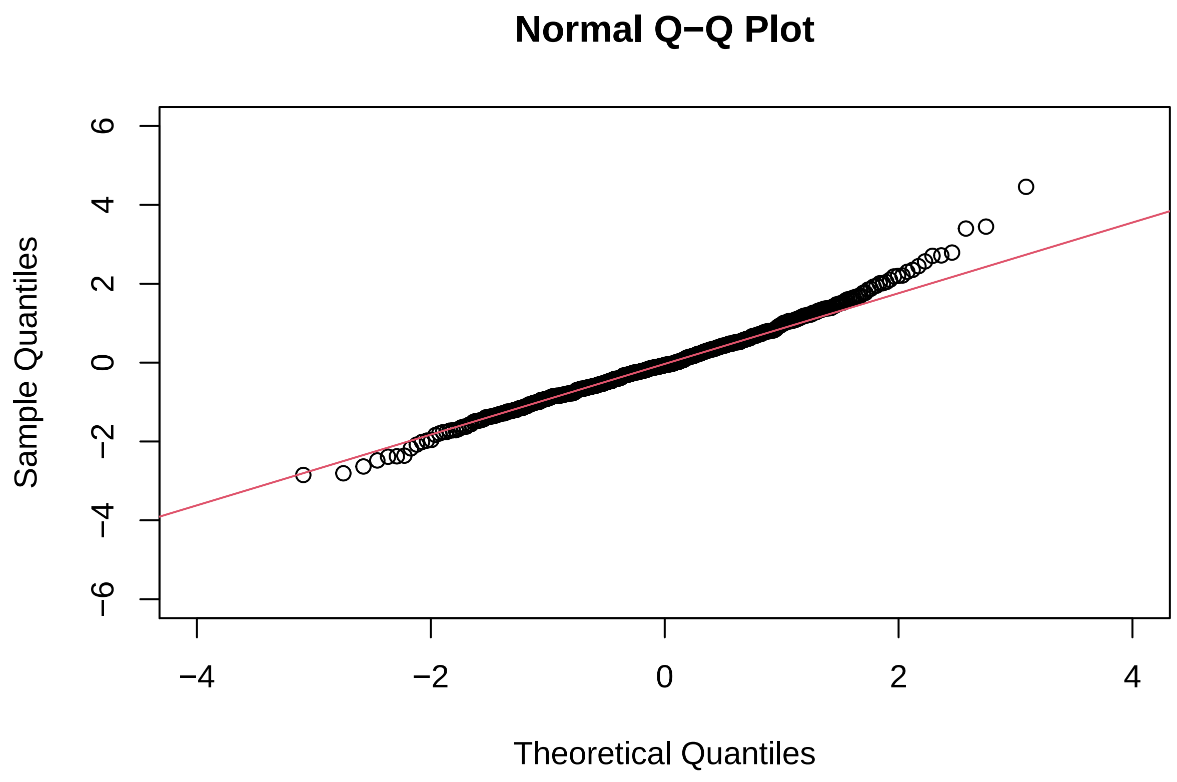


```r
plot(comp1$wgt ~ comp1$WC, xlab = "WC", ylab = "wgt")
```



With a qq plot, we do not find anything disobeys our assumption.

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



```r
pooled_ests <- pool(fit)
summary(pooled_ests, conf.int = TRUE)
```

Step 4: Pooling out the results

```
##           term     estimate  std.error  statistic       df      p.value
## 1  (Intercept) -100.6248327 7.73383679 -13.010985 418.8670 0.000000e+00
## 2 genderfemale   -1.3838520 0.83661318  -1.654112 464.6334 9.878010e-02
## 3          age   -0.1575674 0.02145031  -7.345693 447.6026 9.767742e-13
## 4          hgt   52.3131982 4.43226231  11.802821 411.8402 0.000000e+00
## 5           WC    1.0255694 0.02241459  45.754550 474.6354 0.000000e+00
##          2.5 %      97.5 %
## 1 -115.8267999 -85.4228655
## 2   -3.0278661   0.2601622
## 3   -0.1997232  -0.1154116
## 4   43.6005191  61.0258772
## 5    0.9815253   1.0696135
```

Also, the adjusted R^2 is calculated by:

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

```
##                est     lo 95     hi 95 fmi
## adj R^2 0.8559578 0.8303636 0.8779701 NaN
```

Then we do the Wald test for different variables:

```r
fit_no_gender <- with(imp, lm(wgt ~ age + hgt + WC))
fit_no_age <- with(imp, lm(wgt ~ gender + hgt + WC))
fit_no_hgt <- with(imp, lm(wgt ~ gender + age + WC))
fit_no_WC <- with(imp, lm(wgt ~ gender + age + hgt))

# Wald Test for gender
D1(fit, fit_no_gender)
```

```
##    test statistic df1      df2 dfcom    p.value        riv
##  1 ~~ 2  2.736087   1 478.4232   495 0.09876064 0.03813599
```

```r
# Wald Test for age
D1(fit, fit_no_age)
```

```
##    test statistic df1      df2 dfcom      p.value        riv
##  1 ~~ 2  53.95921   1 464.5474   495 9.263701e-13 0.05509501
```

```r
# Wald test for hgt
D1(fit, fit_no_hgt)
```

```
##    test statistic df1      df2 dfcom p.value        riv
##  1 ~~ 2  139.3066   1 430.9537   495       0 0.08736065
```

```r
# Wald test for WC
D1(fit, fit_no_WC)
```

```
##    test statistic df1      df2 dfcom p.value        riv
##  1 ~~ 2  2093.479   1 485.3969   495       0 0.02698227
```

And we can conclude to see that the Wald test statistic of "gender" is not significant, and therefore the gender has no relevant contribution to the SBP model. However, the Wald test statistics of the other three variables are significant, then these three need to be kept in our model. And the conclusion is same as what we conclude at the plot between response and one variable seperately.