Missing Data

Iuliia Oblasova

11/7/2018

Part 1.

```r
treeage <- read.csv("~/Desktop/702 - Data modeling/treeage.txt")
library(mice)
```

a.

Create a dataset with 30% of the age values missing completely at random, leaving all values of diameter observed.

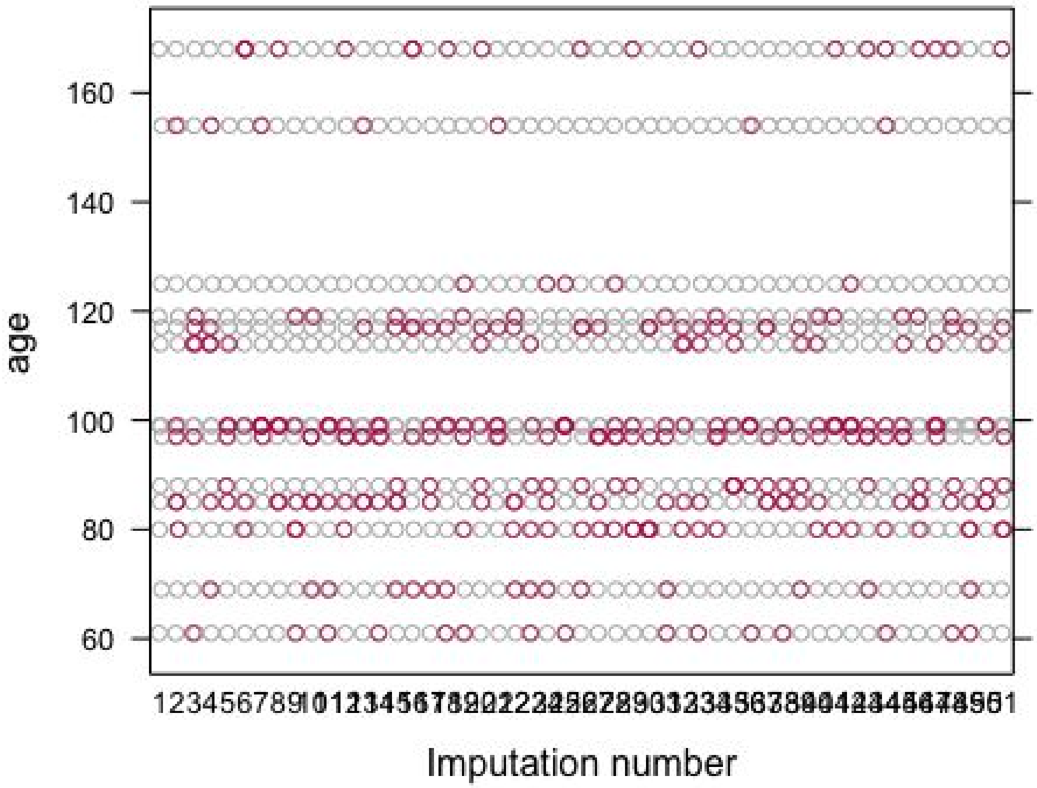
```r
set.seed(12345)
treeage$age[sample(1:20, 6)] = NA
summary(treeage)
```

```
##      number         diameter           age        
##  Min.   : 1.00   Min.   : 5.700   Min.   : 61.00  
##  1st Qu.: 5.75   1st Qu.: 7.975   1st Qu.: 85.75  
##  Median :10.50   Median : 9.250   Median : 99.00  
##  Mean   :10.50   Mean   : 9.405   Mean   :105.36  
##  3rd Qu.:15.25   3rd Qu.:10.850   3rd Qu.:118.50  
##  Max.   :20.00   Max.   :12.000   Max.   :168.00  
##                                   NA's   :6       
```

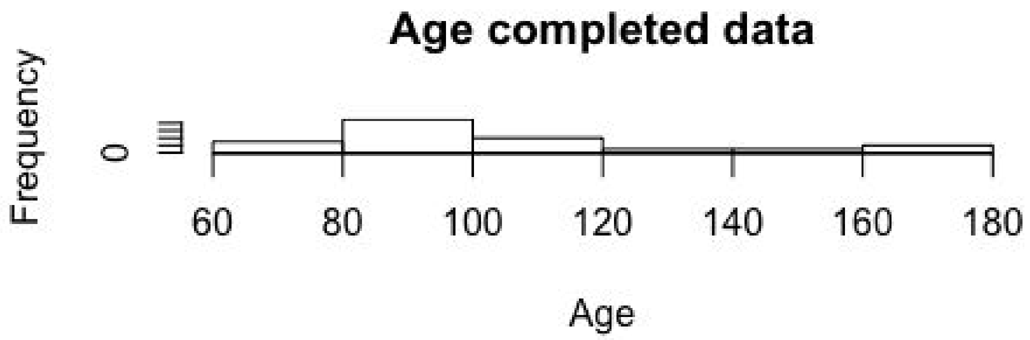
I created a seed = 12345, so the missing data will remain unchanged when generated adain. In this case, the rows 3,8, 14-17 contain missing values for age.

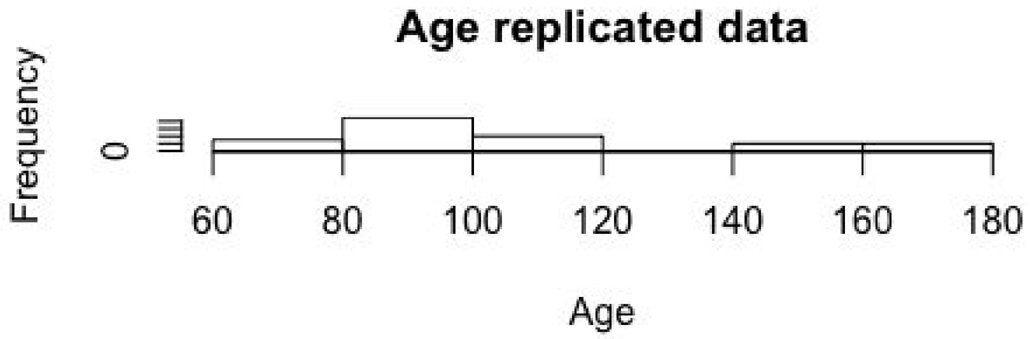
b.

```r
treeMI50 = mice(treeage, m=50)

#look at the first couple of completed datasets
d1 = complete(treeMI50, 1)
d2 = complete(treeMI50, 2)

#Check the imputed values for age.
stripplot(treeMI50, age~.imp, col=c("grey",mdc(2)),pch=c(1,20)))
```

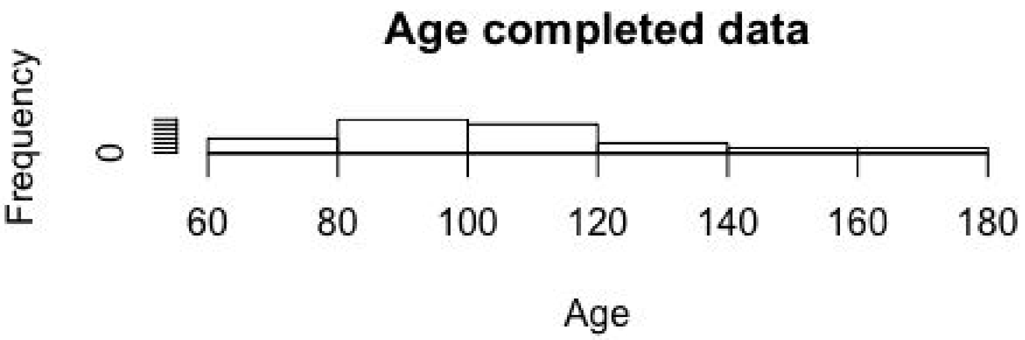


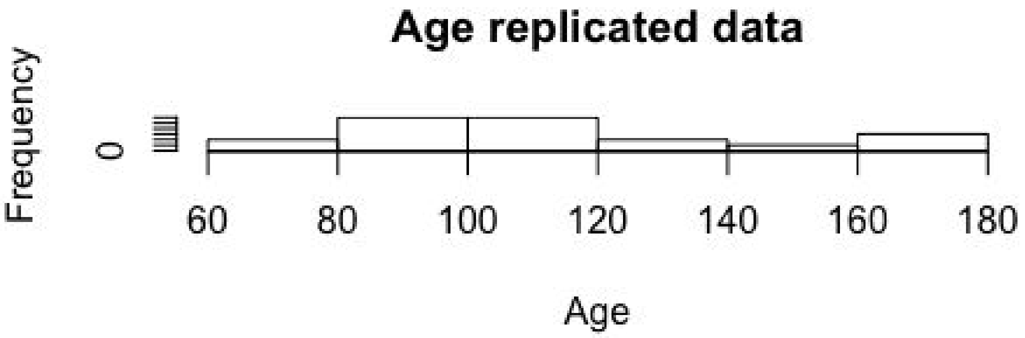
```
#Imputed values look reasonably close.

trees2MI50_bind = rbind(treeage, treeage)
trees2MI50_bind[21:40, 3] = NA
trees2MI50_bind_MI = mice(trees2MI50_bind, m=50)
d1ppcheck = complete(trees2MI50_bind_MI, 1)
d2ppcheck = complete(trees2MI50_bind_MI, 2)

par(mfcol=c(2,1))
hist(d1ppcheck$age[1:20], xlab = "Age", main = "Age completed data")
hist(d1ppcheck$age[21:40], xlab = "Age", main = "Age replicated data")
```



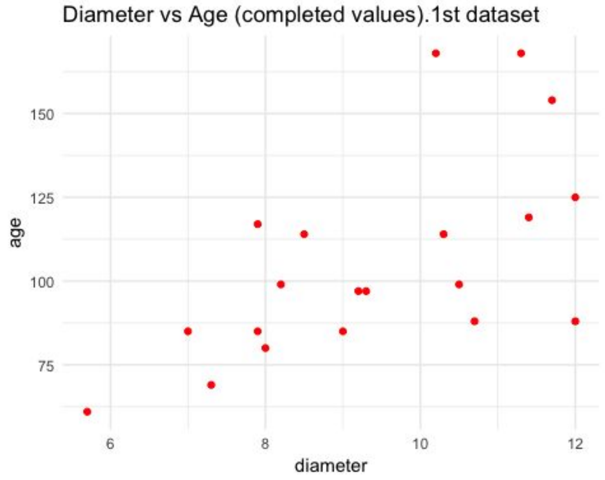


```
par(mfcol=c(2,1))
hist(d2ppcheck$age[1:20], xlab = "Age", main = "Age completed data")
hist(d2ppcheck$age[21:40], xlab = "Age", main = "Age replicated data")
```

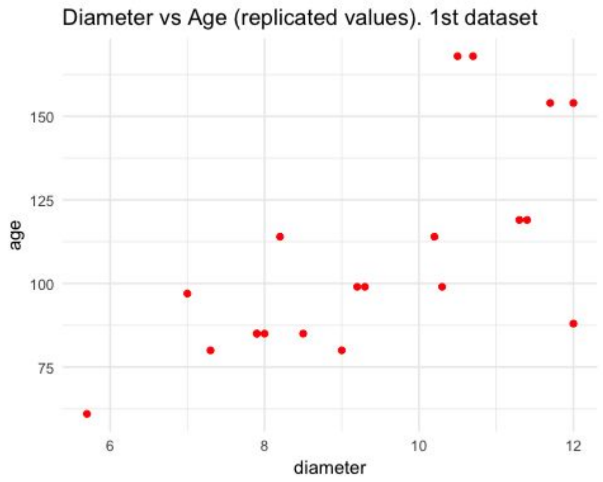




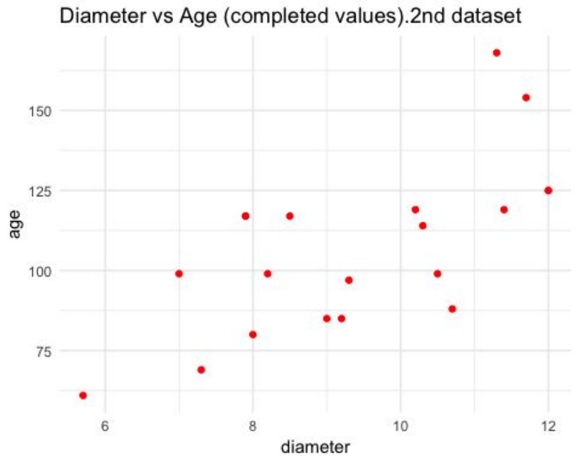
```
#Histograms of the completed and the replicated data look similar, what
suggests that quality of the model meets expectations.

ggplot() + geom_point(data = d1ppcheck[c(1:20),], aes(diameter, age), colour
= 'red')+ theme_minimal() + ggtitle('Diameter vs Age (completed values).1st
dataset')
```



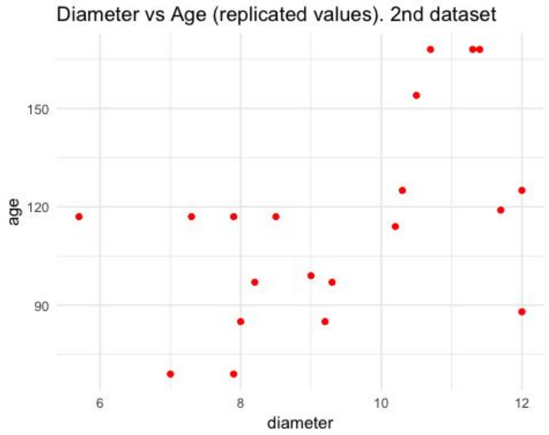
```
ggplot() + geom_point(data = d1ppcheck[c(21:40),], aes(diameter, age), colour
= 'red')+ theme_minimal() + ggtitle('Diameter vs Age (replicated values). 1st
dataset')
```



```
ggplot() + geom_point(data = d2ppcheck[c(1:20),], aes(diameter, age), colour
= 'red')+ theme_minimal() + ggtitle('Diameter vs Age (completed values).2nd
dataset')
```



```
ggplot() + geom_point(data = d2ppcheck[c(21:40),], aes(diameter, age), colour
= 'red')+ theme_minimal() + ggtitle('Diameter vs Age (replicated values). 2nd
dataset')
```

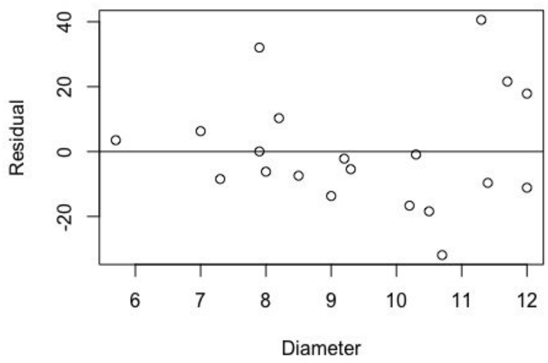


```
#Scatter plots also look similar.
```

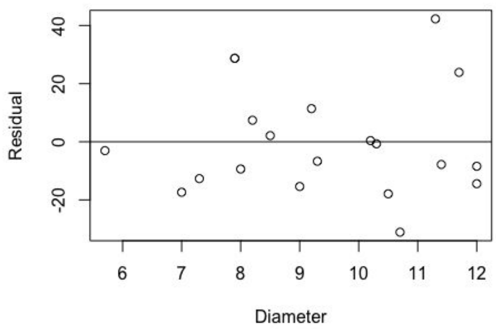
Graphs look similar to the initial data, so no transformation needed and we skip part d).

d).

```
agereg1 = lm(age~diameter, data = d1)

#Check the residuals of the diameter for the first dataset.
plot(agereg1$residual, x = d1$diameter, xlab = "Diameter", ylab = "Residual")
abline(0,0)
```



```
#Check the residuals of the diameter for the second dataset.
agereg2 = lm(age~diameter, data = d2)
plot(agereg2$residual, x = d2$diameter, xlab = "Diameter", ylab = "Residual")
abline(0,0)
```



```
#In both cases residuals look good, however some non-constant variance can be
observed, so, potentially, log transformation could be useful.

ageregMI50 = with(data=treeMI50, lm(age~diameter))
pool(ageregMI50)

## Class: mipo    m = 50
##                 estimate       ubar         b          t dfcom       df
## (Intercept) -0.1513858 615.671508 171.98472 791.095919    18 12.51520
## diameter    11.0546928   6.718994   2.19877   8.961739    18 12.02246
##                    riv    lambda       fmi
## (Intercept) 0.2849318 0.2217486 0.3220698
## diameter    0.3337918 0.2502578 0.3500740
```

```
summary(agereg1, conf.int = T)
```

```
## 
## Call:
## lm(formula = age ~ diameter, data = d1)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -31.920 -10.034  -3.815   7.285  40.588 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  -13.688     21.665  -0.632    0.535    
## diameter      12.487      2.263   5.517 3.07e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 18.04 on 18 degrees of freedom
## Multiple R-squared:  0.6284, Adjusted R-squared:  0.6078 
## F-statistic: 30.44 on 1 and 18 DF,  p-value: 3.074e-05
```

In general, the quality of this simple model is good.

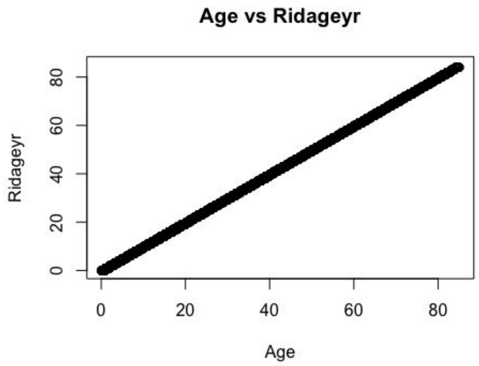
Part 2. Multiple imputation in NHANES data.

```
#Read the file and substitute all dots for the NA's.
nhanes_bmi <- read.csv("~/Desktop/702 - Data modeling/nhanes.csv",
na.strings='.')
nhanes_bmi$wtmec2yr<-NULL
nhanes_bmi$sdmvstra<-NULL
nhanes_bmi$sdmvpsu<-NULL

sum(is.na(nhanes_bmi))/prod(dim(nhanes_bmi))
```

```
## [1] 0.09668708
```

9.67% of the missing values.

```
plot(x=nhanes_bmi$age,y=nhanes_bmi$ridageyr,xlab="Age",ylab="Ridageyr",main="
Age vs Ridageyr")
```



Since age and ridageyr are highly correlated, we drop variable age because it contains missing values.

```
nhanes$age<-NULL
```

Use MICE package for missing values.

```
library(mice)
bmi2MI10 = mice(nhanes, m=10)
```

Let’s check the quality of the imputations for bmi.

#BMI

stripplot(bmi2MI10, bmxbmi~.imp, col=c("grey",mdc(2),pch=c(1,20)))

#Gender

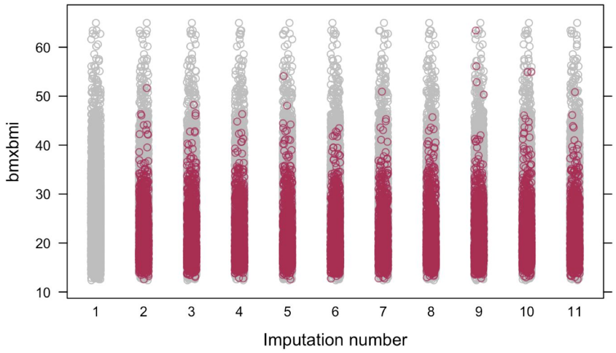
stripplot(bmi2MI10, bmxbmi~ridageyr|riagendr, col=c("grey",mdc(2),pch=c(1,20)))

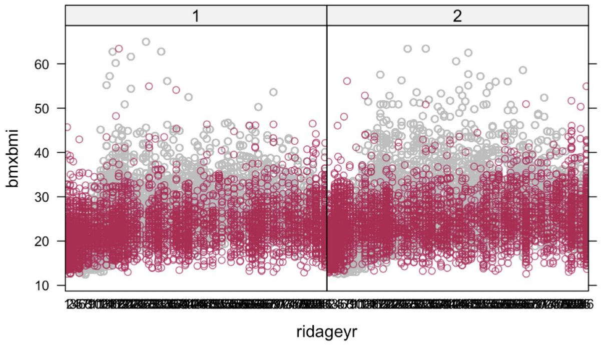
#Education

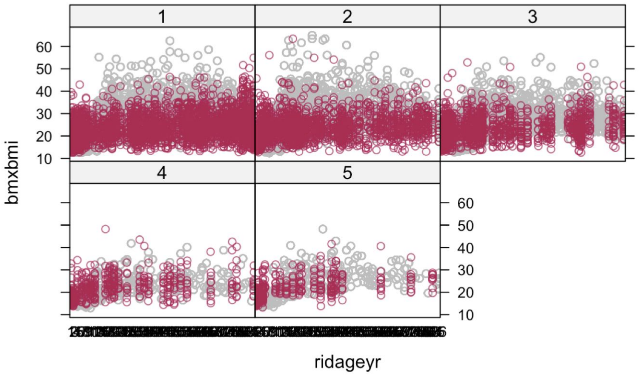
stripplot(bmi2MI10, bmxbmi~ridageyr|dmdeduc, col=c("grey",mdc(2),pch=c(1,20)))

#Race

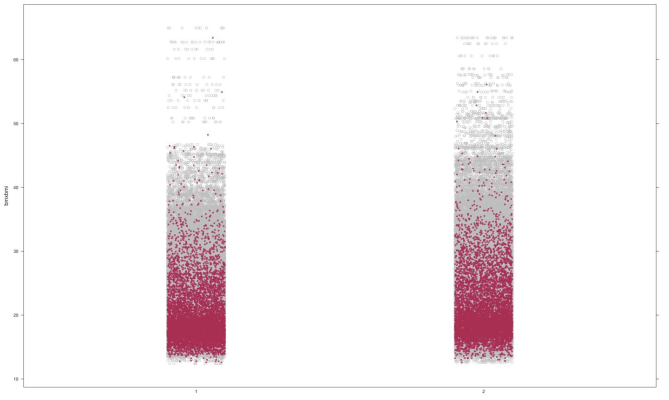
stripplot(bmi2MI10, bmxbmi~ridageyr|ridreth2, col=c("grey",mdc(2),pch=c(1,20)))

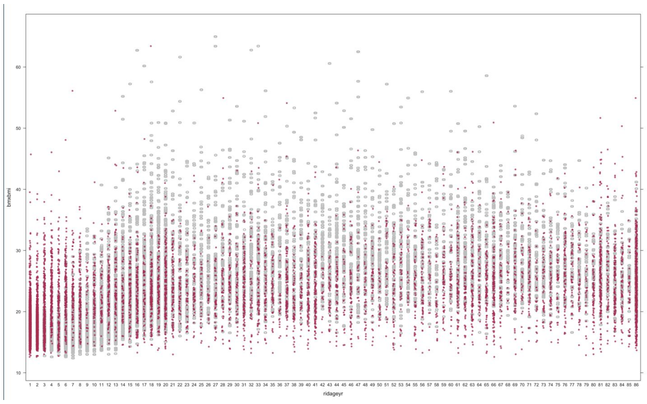






Imputed values look reasonable. Turn in graphical displays for bmxbmi (BMI measurement) by age and bmxbmi by riagendr (gender).

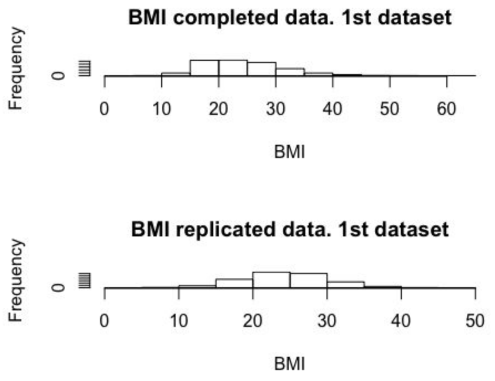




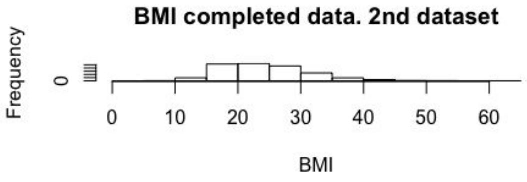
Let’s check the quality of the imputed values.

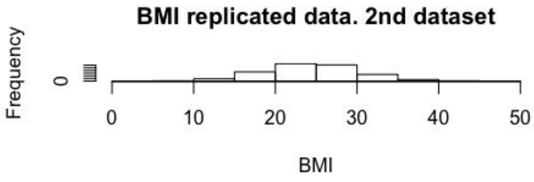
Make the histograms to see of the imputed values fit well.

```
check = rbind(nhanes_bmi, nhanes_bmi)
check[10123:20244, 4:11] = NA
checkMI = mice(check, m=10, defaultMethod = c("norm", "logreg", "polyreg",
"polr"))

check1 = complete(checkMI, 1)
check2 = complete(checkMI, 2)

par(mfcol=c(2,1))
hist(check1$bmxbmi[1:10122], xlab = "BMI", main = "BMI completed data. 1st
dataset")
hist(check1$bmxbmi[10123:20244], xlab = "BMI", main = "BMI replicated data.
1st dataset")
```



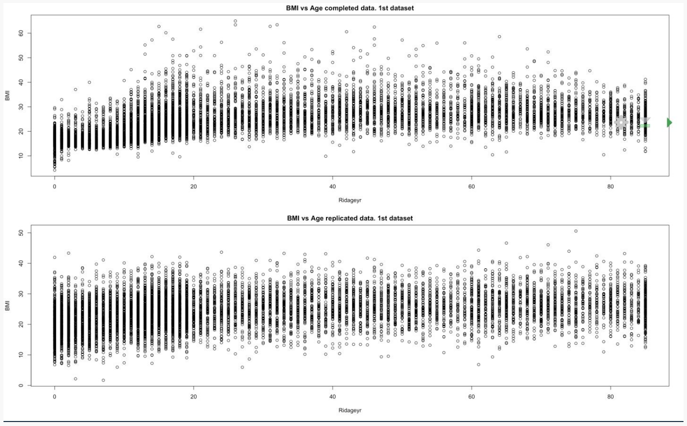
```
hist(check2$bmxbmi[1:10122], xlab = "BMI", main = "BMI completed data. 2nd
dataset")
hist(check2$bmxbmi[10123:20244], xlab = "BMI", main = "BMI replicated data.
2nd dataset")
```

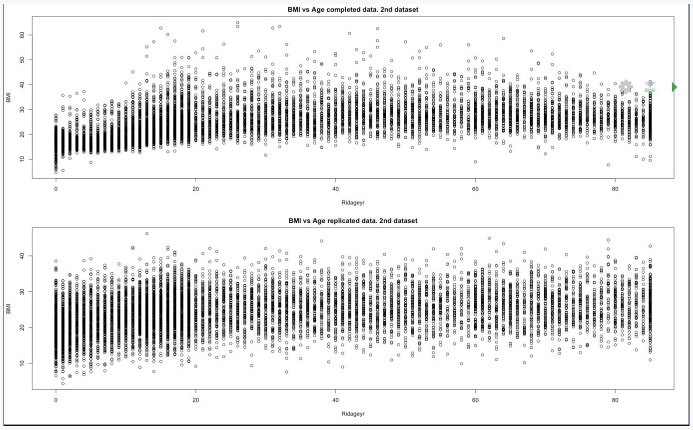




Make scatter plots to check relationship bmi and age.

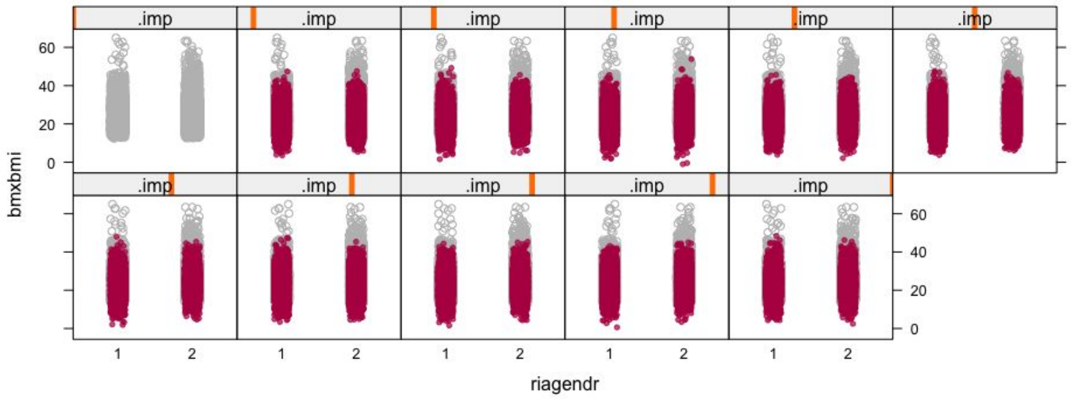
```
par(mfcol=c(2,1))
plot(check1$bmxbmi[1:10122]~check1$ridageyr[1:10122], ylab = "BMI", xlab =
"Ridageyr", main = "BMI vs Age completed data. 1st dataset")
plot(check1$bmxbmi[10123:20244]~check1$ridageyr[10123:20244], ylab = "BMI",
xlab = "Ridageyr", main = "BMI vs Age replicated data. 1st dataset")
```





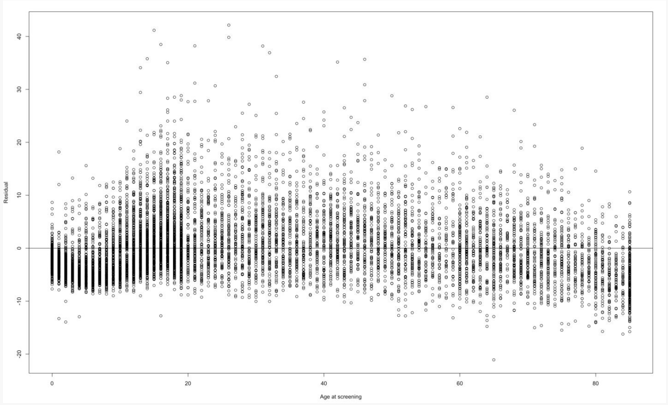
```
plot(check2$bmxbmi[1:10122]~check2$ridageyr[1:10122], ylab = "BMI", xlab =
"Ridageyr", main = "BMI vs Age completed data. 2nd dataset")
plot(check2$bmxbmi[10123:20244]~check2$ridageyr[10123:20244], ylab = "BMI",
xlab = "Ridageyr", main = "BMI vs Age replicated data. 2nd dataset")
```

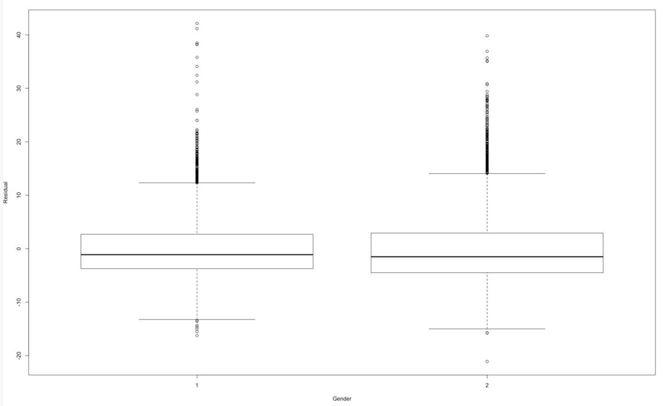
```
#Make  plots to check relationship bmi and gender
stripplot(checkMI, bmxbmi~riagendr|.imp, col=c("grey",mdc(2)),pch=c(1,20))
```

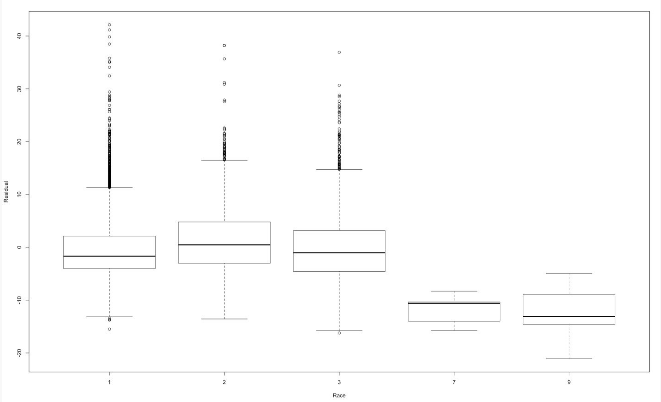


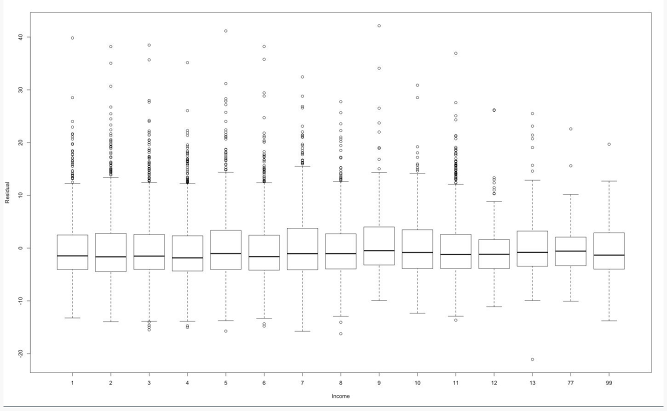
2. Run a model that predicts BMI from some subset of age, gender, race, education, marital status, and income. Apply the multiple imputation combining rules to obtain point and variance estimates for the regression parameters that account for missing data.

```
bmiregd1 = lm(bmxbmi~ridageyr+riagendr+ridreth2+dmdeduc+indfminc, data = d1)

par(mfcol=c(2,2))
plot(bmiregd1$residual, x = d1$ridageyr, xlab = "Age at screening", ylab = "Residual")
abline(0,0)
boxplot(bmiregd1$residual~d1$riagendr, xlab = "Gender", ylab = "Residual")
boxplot(bmiregd1$residual~d1$dmdeduc, xlab = "Race", ylab = "Residual")
boxplot(bmiregd1$residual~d1$indfminc, xlab = "Income", ylab = "Residual")
```

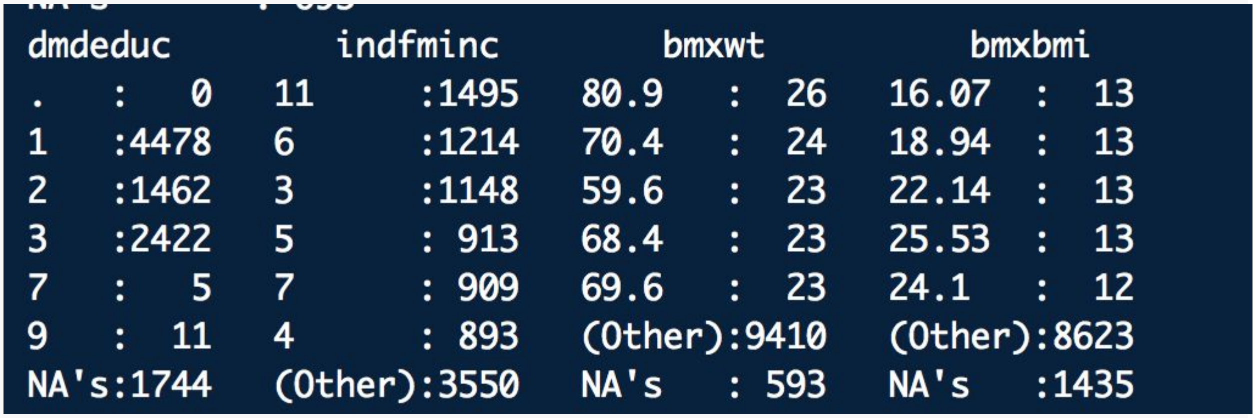








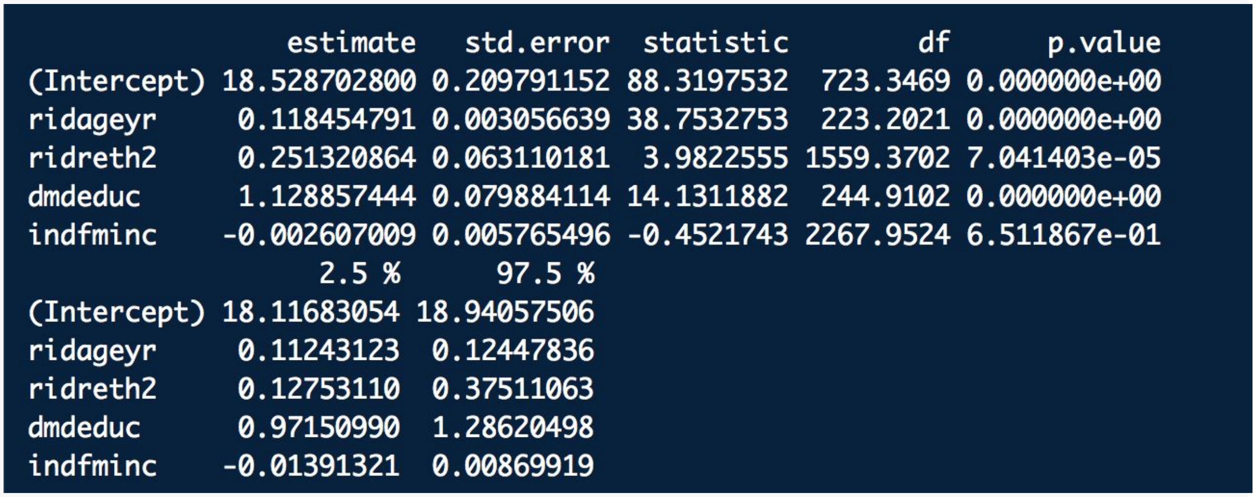
```
finalreg = with(data=nhanes_bmi2MI10,
lm(bmxbmi~ridageyr+ridreth2+dmdeduc+indfminc))
bmireg = pool(finalreg)
summary(bmireg)
```



```
 dmdeduc         indfminc          bmxwt             bmxbmi     
 .    :   0    11      :1495   80.9    :  26    16.07   :  13  
 1    :4478    6       :1214   70.4    :  24    18.94   :  13  
 2    :1462    3       :1148   59.6    :  23    22.14   :  13  
 3    :2422    5       : 913   68.4    :  23    25.53   :  13  
 7    :   5    7       : 909   69.6    :  23    24.1    :  12  
 9    :  11    4       : 893   (Other):9410    (Other):8623  
 NA's:1744    (Other):3550    NA's    : 593    NA's    :1435  
```

Income potentially could be not a significant predictor.

```
finalreg = with(data=nhanes_bmi2MI10,
lm(bmxbmi~ridageyr+ridreth2+dmdeduc+indfminc))
finalreg_noincome = with(data=nhanes_bmi2MI10,
lm(bmxbmi~ridageyr+ridreth2+dmdeduc))
pool.compare(finalreg, finalreg_noincome)
summary(bmireg, conf.int = T)
```



```
                 estimate   std.error  statistic        df      p.value
(Intercept) 18.528702800 0.209791152 88.3197532  723.3469 0.000000e+00
ridageyr     0.118454791 0.003056639 38.7532753  223.2021 0.000000e+00
ridreth2     0.251320864 0.063110181  3.9822555 1559.3702 7.041403e-05
dmdeduc      1.128857444 0.079884114 14.1311882  244.9102 0.000000e+00
indfminc    -0.002607009 0.005765496 -0.4521743 2267.9524 6.511867e-01
                  2.5 %      97.5 %
(Intercept) 18.11683054 18.94057506
ridageyr     0.11243123  0.12447836
ridreth2     0.12753110  0.37511063
dmdeduc      0.97150990  1.28620498
indfminc    -0.01391321  0.00869919
```

P-value = 0.44, so income is not important predictor. Overall, the quality of this model can be improved by transformation; however, the imputed values fall within the expected range.