Missing Data Homework

Guillem_Amat

November 9, 2019

Treeage

This is a dataset on 20 trees comprising age and diameter of trees. Let's create some missing values and run the multiple imputation approach.

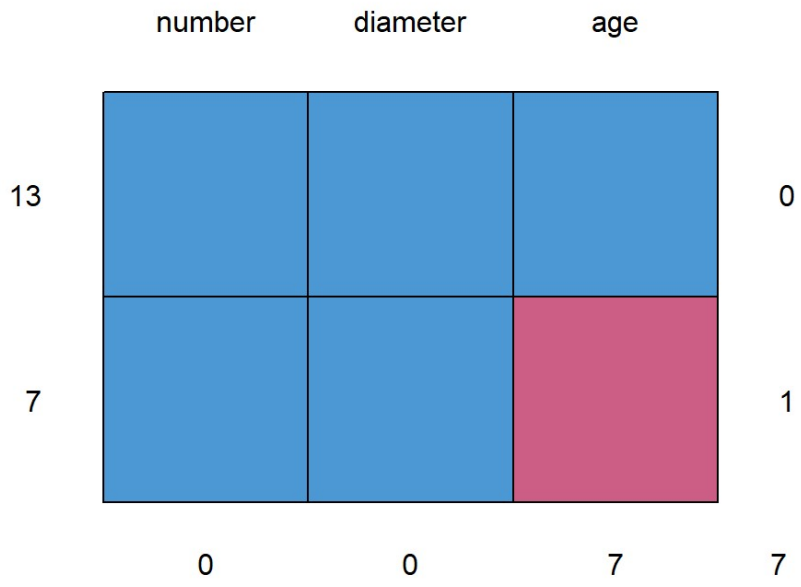
Question 1

Create a dataset with 30% of the age values missing completely at random, leaving all values of diameter observed. Report the R commands you used to make the dataset. Also report the dataset values after you made the ages missing. (This is so we can tell which cases you made missing.)

```
#Set seed
set.seed(123)

#Create extra column with binomial distribution with p = 0.3
TreeData_NA <- cbind(TreeData, rbinom(n = nrow(TreeData), 1, 0.3))
colnames(TreeData_NA)[4] <- "Observations"

#Mark age as NA for Observations = 1
TreeData_NA$age[TreeData_NA$Observations == 1] <- NA
TreeData_NA$Observations <- NULL
TreeData_NA
```

```
##    number diameter age
## 1       1     12.0 125
## 2       2     11.4  NA
## 3       3      7.9  83
## 4       4      9.0  NA
## 5       5     10.5  NA
## 6       6      7.9 117
## 7       7      7.3  69
## 8       8     10.2  NA
## 9       9     11.7 154
## 10     10     11.3 168
## 11     11      5.7  NA
## 12     12      8.0  80
## 13     13     10.3 114
## 14     14     12.0 147
## 15     15      9.2 122
## 16     16      8.5  NA
## 17     17      7.0  82
## 18     18     10.7  88
## 19     19      9.3  97
## 20     20      8.2  NA
```

```
#Visualize missingess patterns
md.pattern(TreeData_NA)
```



```
##    number diameter age
## 13      1        1   1 0
## 7       1        1   0 1
##         0        0   7 7
```

```
md.pairs(TreeData_NA)
```

```
## $rr
##          number diameter age
## number       20       20  13
## diameter     20       20  13
## age          13       13  13
##
## $rm
##          number diameter age
## number        0        0   7
## diameter      0        0   7
## age           0        0   0
##
## $mr
##          number diameter age
## number        0        0   0
## diameter      0        0   0
## age           7        7   0
##
## $mm
##          number diameter age
## number        0        0   0
## diameter      0        0   0
## age           0        0   7
```

Question 2

Use a multiple imputation approach to fill in missing ages with the R software mice using a default application, i.e., no transformations in the imputation models. Create m = 50 imputed datasets.

```
#We will not use number as a
Imputation <- mice(TreeData_NA, print = FALSE)
Prediction_Matrix <- Imputation$predictorMatrix
Prediction_Matrix[, "number"] <- 0

#Imputing the missing values by using a the Normal Method
TreeData_I <- mice(TreeData_NA, m = 50, defaultMethod = "norm", print=F, pred = Pre
diction_Matrix)
TreeData_I
```

```
## Class: mids
## Number of multiple imputations:  50
## Imputation methods:
##   number diameter      age
##       ""       ""   "norm"
## PredictorMatrix:
##          number diameter age
## number        0        1   1
## diameter      0        0   1
## age           0        1   0
```

```
#Checking the value of the first imputation vs the Original Data
T1 <- mice::complete(TreeData_I, 1); T1
```

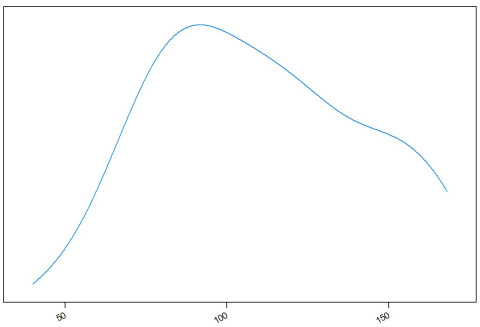
```
##    number diameter       age
## 1       1     12.0 125.00000
## 2       2     11.4 161.39647
## 3       3      7.9  83.00000
## 4       4      9.0 103.85772
## 5       5     10.5  73.07475
## 6       6      7.9 117.00000
## 7       7      7.3  69.00000
## 8       8     10.2 157.17461
## 9       9     11.7 154.00000
## 10     10     11.3 168.00000
## 11     11      5.7  39.99840
## 12     12      8.0  80.00000
## 13     13     10.3 114.00000
## 14     14     12.0 147.00000
## 15     15      9.2 122.00000
## 16     16      8.5  90.64374
## 17     17      7.0  82.00000
## 18     18     10.7  88.00000
## 19     19      9.3  97.00000
## 20     20      8.2 126.53123
```

```
T50 <- mice::complete(TreeData_I, 50); T50
```

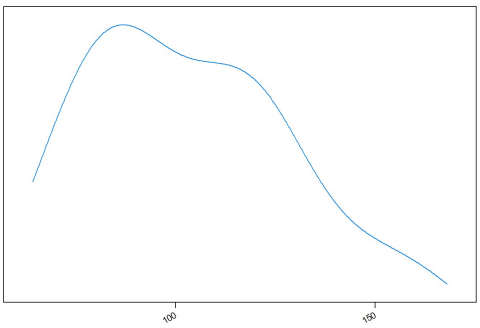
```
##    number diameter       age
## 1       1     12.0 125.00000
## 2       2     11.4 115.55918
## 3       3      7.9  83.00000
## 4       4      9.0  64.18967
## 5       5     10.5 126.63896
## 6       6      7.9 117.00000
## 7       7      7.3  69.00000
## 8       8     10.2 110.59317
## 9       9     11.7 154.00000
## 10     10     11.3 168.00000
## 11     11      5.7  90.83390
## 12     12      8.0  80.00000
## 13     13     10.3 114.00000
## 14     14     12.0 147.00000
## 15     15      9.2 122.00000
## 16     16      8.5  78.33118
## 17     17      7.0  82.00000
## 18     18     10.7  88.00000
## 19     19      9.3  97.00000
## 20     20      8.2  89.16061
```

Use multiple imputation diagnostics to check the quality of the imputations of age, looking at both the marginal distribution of age and the scatter plot of age versus diameter. Run the diagnostics on at least two of the completed datasets. Turn in the graphical displays you made (showing results for at least two completed datasets) and your conclusions about the quality of the imputation model.

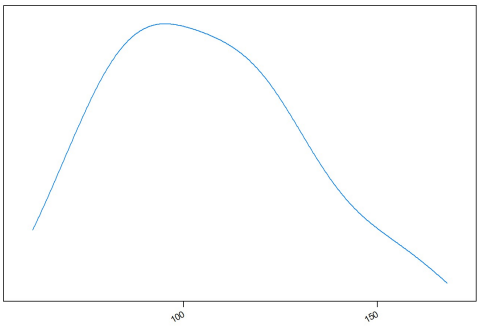
```
#Marginal plot
par(mfrow = c(1,3))
marginal.plot(T1$age)
```



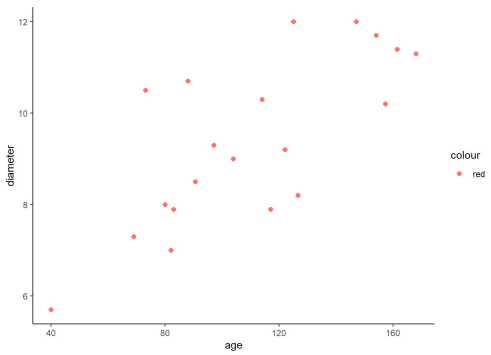

```
marginal.plot(T50$age)
```



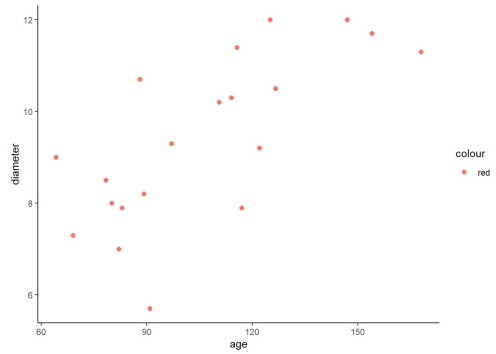

```
marginal.plot(TreeData$age)
```



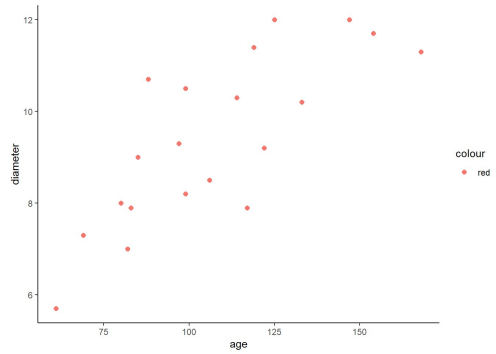

```
#Age vs Diameter Scatter Plot
par(mfrow = c(1,3))

ggplot(T1, aes(x=age, y=diameter, col = "red")) +
  geom_point(size=2) + theme_classic()
```



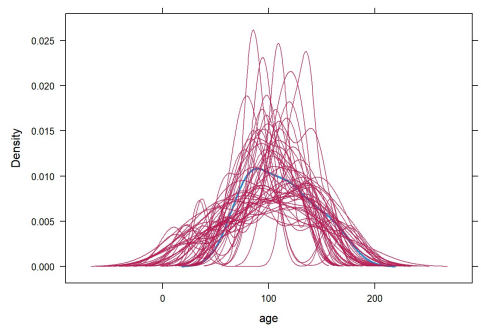

```
ggplot(T50, aes(x=age, y=diameter, col = "red")) +
  geom_point(size=2) + theme_classic()
```



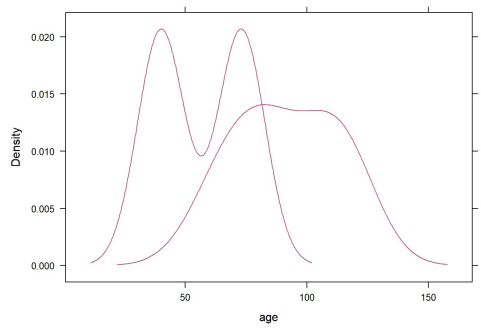

```
ggplot(TreeData, aes(x=age, y=diameter, col = "red")) +
  geom_point(size=2) + theme_classic()
```




```
#Density Plot
densityplot(TreeData_I)
```

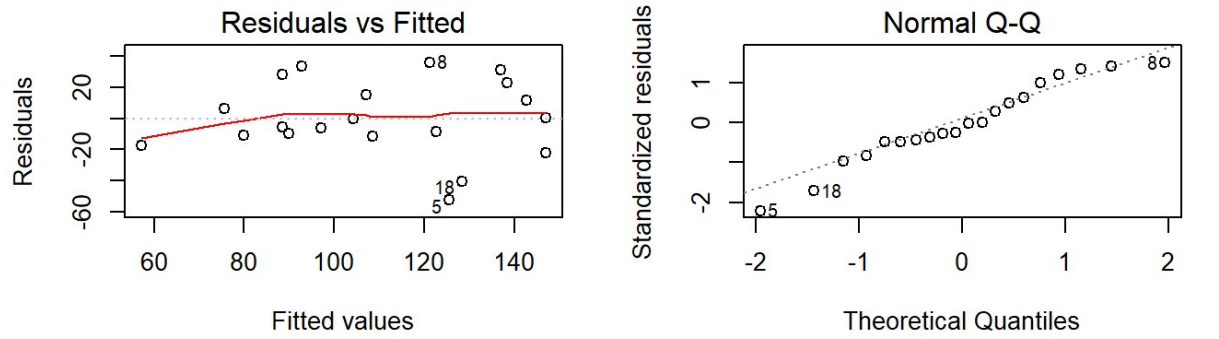



```
densityplot(TreeData_I, subset = .imp ==c(1,50))
```

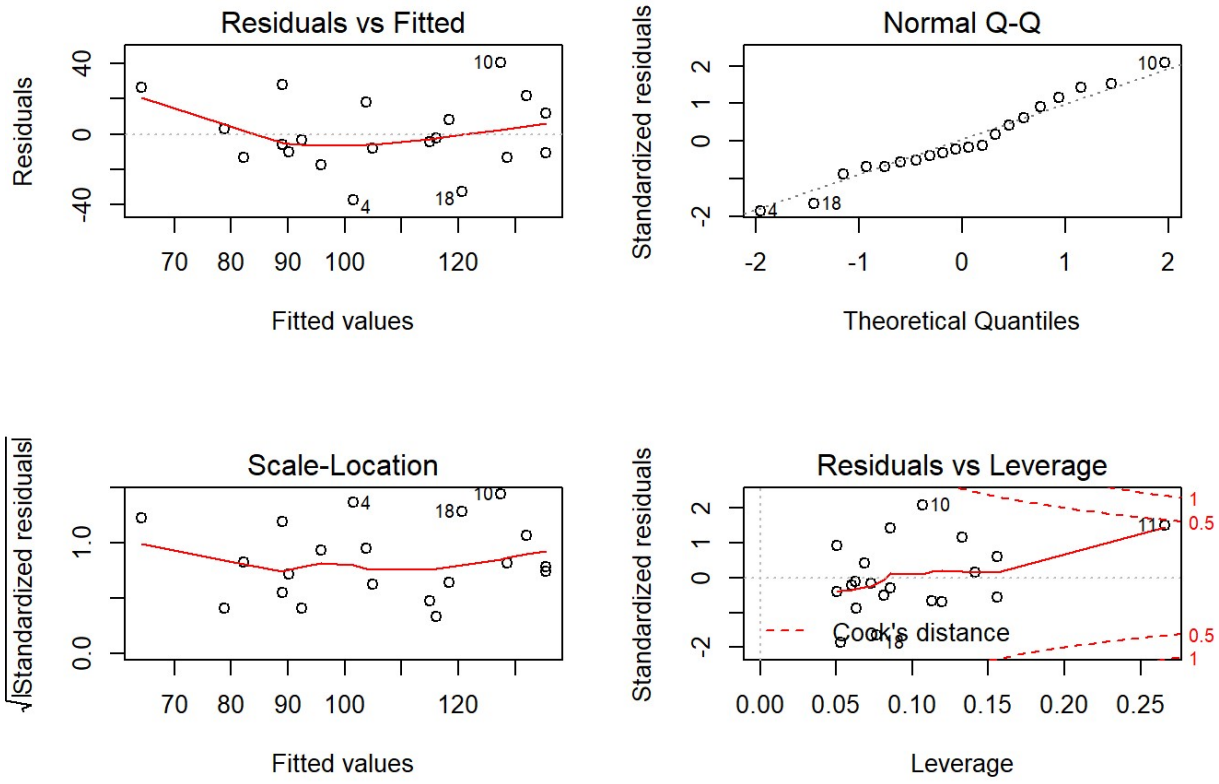


From the scatter plots it seems that the two imputed datasets are linear enough. The marginal plots seem to adhere to normality for T1 but not for T50. We will fit the models to the data and check for the different assumptions.

```
# Fitting a model to the Imputation data of T1 and T50
T1_Model <- lm(age ~ diameter, T1)
T50_Model <- lm(age ~ diameter, T50)

#Assumptions for T1
par(mfrow = c(2,2))
plot(T1_Model, which = 1); plot(T1_Model, which = 2); plot(T1_Model, which = 3); pl
ot(T1_Model, which = 5)
```




```
#Assumptions for T50
par(mfrow = c(2,2))
plot(T50_Model, which = 1); plot(T50_Model, which = 2); plot(T50_Model, which = 3);
plot(T50_Model, which = 5)
```



We do not observe any major issues with model assumptions. We observe a little bit of skew in the T50 normal Q-Q plot and a bit of uneveness in Variance for both T1 and T50, but that is mostly due to the small sample size (20 values). We will therefore not apply any transformations to the underlying data.

```
Tree_Pooled <- with(TreeData_I, lm(age ~ diameter))

#I have commented the summary line because the output is way too long
#summary(Tree_Pooled)
```

Conclusions

The quality of imputations is quite good. After comparing the models of the Full dataset to a couple of the imputations we can see that the slope for diameter is pretty accurate. The Intercept is a bit far from the real value in the T50 imputed dataset for instance, but that could also be due to the small number of observations of the dataset. Overall, the imputation method worked quite well and our model would be close enough to the real thing

Nhanes

```
#Importing the data
Nhanes_Data <- read.csv("nhanes.csv", header = TRUE, sep = ",")

#Changing the type of factor columns to numeric
Nhanes_Data$age <- as.numeric(Nhanes_Data$age)
Nhanes_Data$riagendr <- as.factor(Nhanes_Data$riagendr)
Nhanes_Data$ridreth2 <- as.factor(Nhanes_Data$ridreth2)
Nhanes_Data$bmxwt <- as.numeric(Nhanes_Data$bmxwt)
Nhanes_Data$bmxbmi <- as.numeric(Nhanes_Data$bmxbmi)
Nhanes_Data$bmxtri <- as.numeric(Nhanes_Data$bmxtri)
Nhanes_Data$bmxwaist <- as.numeric(Nhanes_Data$bmxwaist)
Nhanes_Data$bmxthicr <- as.numeric(Nhanes_Data$bmxthicr)
Nhanes_Data$bmxarml <- as.numeric(Nhanes_Data$bmxarml)
Nhanes_Data$bmxarml <- as.numeric(Nhanes_Data$bmxarml)


Nhanes_Data$dmdeduc <- as.numeric(Nhanes_Data$dmdeduc) #Converting as num
Nhanes_Data$dmdeduc <- as.factor(Nhanes_Data$dmdeduc) #Refactorizing to drop unused
levels

Nhanes_Data$indfminc <- as.numeric(Nhanes_Data$indfminc)
Nhanes_Data$indfminc <- as.factor(Nhanes_Data$indfminc)

#Changing dots(missing values) in the data to NAs
Nhanes_Data[Nhanes_Data == "."] <- NA

#Dropping unecessary variables
Nhanes_Data <- Nhanes_Data %>%
  select(-c("sdmvstra", "sdmvpsu", "wtmec2yr"))

#Cheking the data
str(Nhanes_Data)
```

```
## 'data.frame':    10122 obs. of  12 variables:
##  $ age     : num  139 111 78 115 626 588 6 731 70 284 ...
##  $ ridageyr: int  19 16 14 17 55 52 0 63 13 3 ...
##  $ riagendr: Factor w/ 2 levels "1","2": 1 2 2 1 1 2 1 1 2 1 ...
##  $ ridreth2: Factor w/ 5 levels "1","2","3","4",..: 2 2 1 2 1 1 3 2 2 2 ...
##  $ dmdeduc : Factor w/ 6 levels "1","2","3","4",..: 2 2 2 2 3 4 1 3 2 1 ...
##  $ indfminc: Factor w/ 16 levels "1","2","3","4",..: 11 11 11 12 14 9 6 7 7 9
...
##  $ bmxwt   : num  312 814 731 982 43 951 868 854 809 284 ...
##  $ bmxbmi  : num  2584 708 477 695 1741 ...
##  $ bmxtri  : num  1 92 361 358 90 173 16 356 58 358 ...
##  $ bmxwaist: num  340 719 678 730 186 897 1 720 718 467 ...
##  $ bmxthicr: num  458 174 117 190 255 171 1 1 184 1 ...
##  $ bmxarml : num  311 228 227 291 322 250 7 261 207 93 ...
```

Question 1

Use a multiple imputation approach to fill in missing values with the R software mice using a default application (no transformations in the modeling).

```
#Using mice to create 10 imputations
Nhanes_I <- mice(Nhanes_Data, m = 10, defaultMethod = "norm", print=F)
Nhanes_I
```

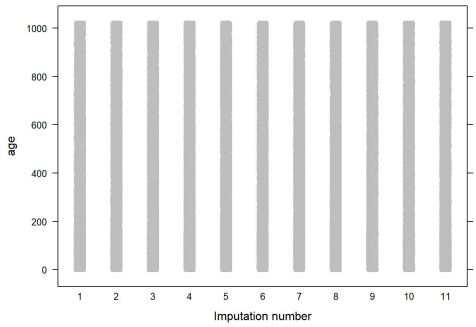
```
## Class: mids
## Number of multiple imputations:  10
## Imputation methods:
##      age ridageyr riagendr ridreth2  dmdeduc indfminc    bmxwt   bmxbmi
##       ""       ""       ""       ""       ""       ""       ""       ""
##   bmxtri bmxwaist bmxthicr  bmxarml
##       ""       ""       ""       ""
## PredictorMatrix:
##          age ridageyr riagendr ridreth2 dmdeduc indfminc bmxwt bmxbmi
## age        0        1        1        1       1        1     1      1
## ridageyr   1        0        1        1       1        1     1      1
## riagendr   1        1        0        1       1        1     1      1
## ridreth2   1        1        1        0       1        1     1      1
## dmdeduc    1        1        1        1       0        1     1      1
## indfminc   1        1        1        1       1        0     1      1
##          bmxtri bmxwaist bmxthicr bmxarml
## age           1        1        1       1
## ridageyr      1        1        1       1
## riagendr      1        1        1       1
## ridreth2      1        1        1       1
## dmdeduc       1        1        1       1
## indfminc      1        1        1       1
```

```
#Checking the first and the last imputation
N1 <- mice::complete(Nhanes_I, 1); #N1
N10 <- mice::complete(Nhanes_I, 10); #N10
```

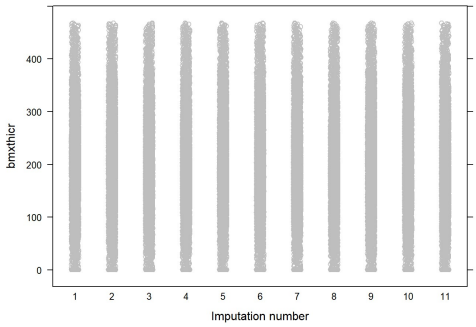
Question 2

Use multiple imputation diagnostics to check the quality of the imputations, looking at both marginal distributions and scatter plots. Run the diagnostics on at least two of the completed datasets. Turn in plots for bmxbmi (BMI measurement) by age and bmxbmi by riagendr (gender)

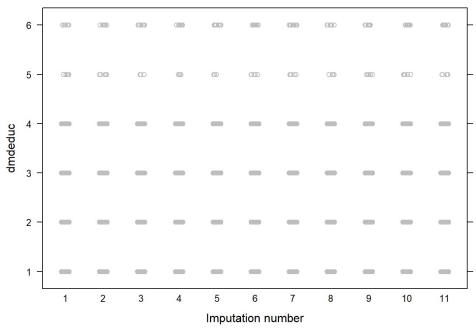
```
par(mfrow = c(2,2))

stripplot(Nhanes_I, age ~ .imp, col = c('grey', 'darkred'), pch = c(1, 1))
```



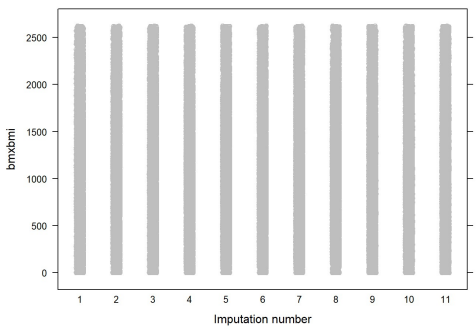

```
stripplot(Nhanes_I, bmxthicr ~ .imp, col = c('grey', 'darkred'), pch = c(1, 1))
```



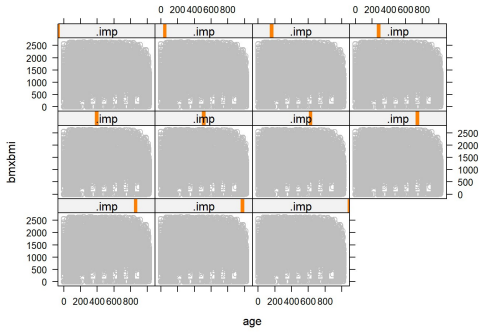

```
stripplot(Nhanes_I, dmdeduc ~ .imp, col = c('grey', 'darkred'), pch = c(1, 1))
```



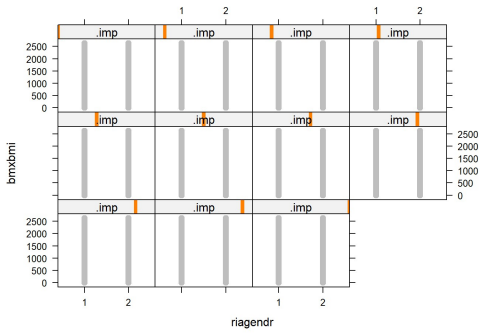

```
stripplot(Nhanes_I, bmxbmi ~ .imp, col = c('grey', 'darkred'), pch = c(1, 1))
```




```
#Plotting bmxbmi vs age and bmxbmi vs riagendr
xyplot(Nhanes_I, bmxbmi ~ age | .imp, pch = c(1, 2), cex = 1, col = c('grey', 'dark
red'))
```




```
xyplot(Nhanes_I, bmxbmi ~ riagendr | .imp, pch = c(1, 2), cex = 1, col = c('grey',
'darkred'))
```



We can conclude from the multiple plots that the model does a pretty good job imputing the values. Most of the imputations are fit in the range of values of the data

Question 3

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

```
Null_model = lm(bmxbmi ~ 1, N1)
Full_model = lm(bmxbmi ~ age + ridageyr + riagendr + ridreth2 + dmdeduc + indfminc,
N1)
Nhanes_Model <- step(Null_model, scope = formula(Full_model), direction = 'forwar
d', trace = 0)
summary(Nhanes_Model)
```

```
##
## Call:
## lm(formula = bmxbmi ~ dmdeduc + ridageyr + ridreth2 + age + riagendr +
##     indfminc, data = N1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1671.16  -322.56   -45.73   306.95  1794.71
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -141.98789   47.04076  -3.018 0.002547 **
## dmdeduc2     653.17301   17.30549  37.744  < 2e-16 ***
## dmdeduc3     909.02723   23.39504  38.856  < 2e-16 ***
## dmdeduc4     897.54583   21.94462  40.901  < 2e-16 ***
## dmdeduc5     225.65791  250.02378   0.903 0.366789
## dmdeduc6      63.49160  169.22765   0.375 0.707531
## ridageyr       5.30121    0.32277  16.424  < 2e-16 ***
## ridreth22    123.85629   14.49784   8.543  < 2e-16 ***
## ridreth23    124.38332   15.07378   8.252  < 2e-16 ***
## ridreth24   -142.52676   31.75235  -4.489 7.25e-06 ***
## ridreth25     80.33563   31.63517   2.539 0.011118 *
## age            0.16001    0.02106   7.597 3.29e-14 ***
## riagendr2     36.36380   11.06133   3.287 0.001014 **
## indfminc2    163.46000   49.49610   3.302 0.000962 ***
## indfminc3    172.02292   52.77737   3.259 0.001120 **
## indfminc4    152.73285   46.72678   3.269 0.001084 **
## indfminc5     84.96405   66.15812   1.284 0.199081
## indfminc6    164.39917   65.06141   2.527 0.011525 *
## indfminc7    177.77132   48.83078   3.641 0.000273 ***
## indfminc8    179.43388   47.26633   3.796 0.000148 ***
## indfminc9    126.94633   48.01611   2.644 0.008210 **
## indfminc10   178.03416   47.94918   3.713 0.000206 ***
## indfminc11   125.35721   47.05642   2.664 0.007735 **
## indfminc12   191.52948   47.98446   3.991 6.61e-05 ***
## indfminc13    73.56364   75.11882   0.979 0.327457
## indfminc14   157.71424   48.73239   3.236 0.001215 **
## indfminc15   218.62826   51.23472   4.267 2.00e-05 ***
## indfminc16   175.51150   76.02037   2.309 0.020978 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 555.4 on 10094 degrees of freedom
## Multiple R-squared:  0.3676, Adjusted R-squared:  0.366
## F-statistic: 217.4 on 27 and 10094 DF,  p-value: < 2.2e-16
```

Almost all the coefficient in the model are very significant. ridreth and dmdeduc are the most significant predictors. I used a forward selection model, starting with a model that had no variables to a complete model with all the meaningful variables but no interactions. R-Squared suggests that 36% of the variability in the data is explained by the model

```
Nhanes_Pooled <- with(Nhanes_I, lm(bmxbmi ~ ridageyr + age + dmdeduc + riagendr + r
idreth2 +
                                    indfminc))

summary(pool(Nhanes_Pooled)) %>%
  kable() %>%
  kable_styling()
```

	estimate	std.error	statistic	df	p.value
(Intercept)	-141.9878873	47.0407648	-3.0184009	10090.88	0.0025475
ridageyr	5.3012068	0.3227681	16.4241984	10090.88	0.0000000
age	0.1600098	0.0210614	7.5973086	10090.88	0.0000000
dmdeduc2	653.1730127	17.3054867	37.7436950	10090.88	0.0000000
dmdeduc3	909.0272292	23.3950427	38.8555490	10090.88	0.0000000
dmdeduc4	897.5458297	21.9446166	40.9005017	10090.88	0.0000000
dmdeduc5	225.6579076	250.0237786	0.9025458	10090.88	0.3667886
dmdeduc6	63.4915993	169.2276489	0.3751845	10090.88	0.7075311
riagendr2	36.3637968	11.0613329	3.2874697	10090.88	0.0010144
ridreth22	123.8562912	14.4978410	8.5430852	10090.88	0.0000000
ridreth23	124.3833200	15.0737836	8.2516323	10090.88	0.0000000
ridreth24	-142.5267567	31.7523484	-4.4886997	10090.88	0.0000072
ridreth25	80.3356257	31.6351687	2.5394404	10090.88	0.0111179
indfminc2	163.4600007	49.4960978	3.3024826	10090.88	0.0009617
indfminc3	172.0229227	52.7773688	3.2594069	10090.88	0.0011202
indfminc4	152.7328463	46.7267822	3.2686361	10090.88	0.0010843
indfminc5	84.9640543	66.1581228	1.2842573	10090.88	0.1990814
indfminc6	164.3991702	65.0614096	2.5268307	10090.88	0.0115248
indfminc7	177.7713235	48.8307791	3.6405588	10090.88	0.0002734
indfminc8	179.4338772	47.2663291	3.7962304	10090.88	0.0001478
indfminc9	126.9463349	48.0161146	2.6438277	10090.88	0.0082101

	estimate	std.error	statistic	df	p.value
indfminc10	178.0341572	47.9491844	3.7129757	10090.88	0.0002059
indfminc11	125.3572149	47.0564196	2.6639769	10090.88	0.0077346
indfminc12	191.5294797	47.9844587	3.9914898	10090.88	0.0000661
indfminc13	73.5636379	75.1188221	0.9792970	10090.88	0.3274567
indfminc14	157.7142408	48.7323897	3.2363330	10090.88	0.0012147
indfminc15	218.6282551	51.2347218	4.2671893	10090.88	0.0000200
indfminc16	175.5114980	76.0203697	2.3087430	10090.88	0.0209780