Final Project

Missing Data

Liuyang Xu

3/13/2022

Library

```
library(tidyverse)
library(VIM)
library(mice)
library(mi)
```

Introduction

Step 0

Find a suitable data set with missing observations. Ideally, it should have at least 100 observations, and at least 3-4 variables, both numerical and categorical, of which at least one numerical variable is completely observed. Decide what model you want to run, what you want to estimate, and which variable you want to predict by the rest.

The data set I chose is called the 'earnings'. This 'earnings' data set comes from a national survey from 1990 and contains the women and men's earnings and other information such as weight and education status. The original data "earnings.csv" is provided in the folder.

I chose "height", "weight", "male", "education", and "earn" from the data set for my model of interest and called this subset as "earnings". This data set contains 5 variables and 1787 observations in total. The "height", "weight", "education", and "earn" are numerical variables and "male" is a categorical variable.

After reading the data, it is stored in "earnings_original" and contains no missing value. I use ampute() from mice package to create missingness with the following command:

```
set.seed(1234)
# read and modify the data
earnings_original <- read_csv("earnings.csv") %>%
  select(height, weight, male, education, earn) %>%
  na.omit()
```

```
# create enough missing value
ans.miss <- ampute(earnings_original[,1:4], prop = 0.24)$amp
earnings <- cbind(ans.miss[, 1:4],earnings_original$earn)
colnames(earnings)[5] = "earn"
```

```r
knitr::kable(head(earnings)) # present the head of the data
```

height	weight	male	education	earn
74	210	1	16	50000
66	125	0	NA	60000
64	126	0	16	30000
65	200	0	17	25000
63	110	0	16	50000
68	165	0	18	62000

The table above shows the head of the data. The numerical variable "earn" is completely observed.

I want to do a linear regression with the data and to find the relationship between the earnings 'earn' and the gender 'male', 'height', 'weight', and education status 'education'. The expected estimated equation is

$$earn = b_0 + b_1 \times height + b_2 \times weight + b_3 \times male + b_4 \times education$$

Main Part

Step 1

Provide some plots and summary statistics, like percent missing per variable, percent complete cases, and so on

The summary of the data set "earnings" is shown below.

```r
summary(earnings)
```

```
##      height          weight           male          education    
##  Min.   :57.00   Min.   : 80.0   Min.   :0.0000   Min.   : 2.00  
##  1st Qu.:64.00   1st Qu.:130.0   1st Qu.:0.0000   1st Qu.:12.00  
##  Median :66.00   Median :150.0   Median :0.0000   Median :12.00  
##  Mean   :66.53   Mean   :155.2   Mean   :0.3608   Mean   :13.23  
##  3rd Qu.:69.00   3rd Qu.:175.0   3rd Qu.:1.0000   3rd Qu.:15.00  
##  Max.   :82.00   Max.   :342.0   Max.   :1.0000   Max.   :18.00  
##  NA's   :104     NA's   :107     NA's   :110      NA's   :124    
##       earn       
##  Min.   :     0  
##  1st Qu.:  6000  
##  Median : 16000  
##  Mean   : 21248  
##  3rd Qu.: 27000  
##  Max.   :400000  
## 
```

Now let's check the missing percent of the dataset.

```r
# What percent of cases is incomplete
missing_percent.total <- 1-sum(complete.cases(earnings))/nrow(earnings)
missing_percent.total
```

```
## [1] 0.2490207
```

```r
missing_percent.each <- colMeans(apply(earnings, 2, is.na))
missing_percent.each
```

```
##     height     weight       male  education       earn
## 0.05819810 0.05987689 0.06155568 0.06939004 0.00000000
```

```r
# What percent of cases is complete
notmissing_percent.total <- sum(complete.cases(earnings))/nrow(earnings)
notmissing_percent.total
```

```
## [1] 0.7509793
```

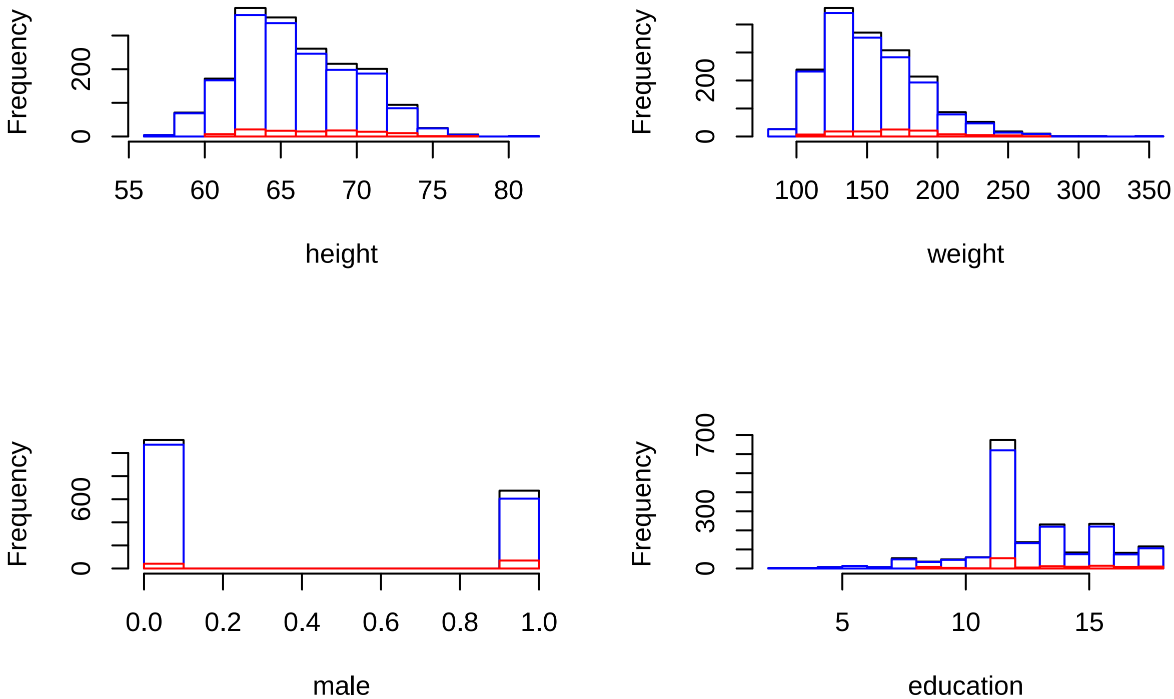
```r
notmissing_percent.each <- 1-colMeans(apply(earnings, 2, is.na))
notmissing_percent.each
```

```
##    height    weight      male education      earn
## 0.9418019 0.9401231 0.9384443 0.9306100 1.0000000
```

There are 24.90207% of the data missing in the data set. Specifically, there are 5.819810% height data missing, 5.987689% weight data missing, 6.155568% gender data missing, and 6.939004% education status missing.
Relatively, there are 75.09793% of the complete data in the data set. Specifically, there are 94.18019% complete height data, 94.01231% complete weight data, 93.84443% complete gender data, and 93.06100% complete education status data.

Now let's show the histogram of these 4 variables.

```r
par(mfrow=c(2,2))
xlabnames <- c("height", "weight", "male", "education")
for (i in 1:4) {
  # Plot "original", "observed",and "missing"
  hist(as.matrix(earnings_original)[ , i],
       col = "white", border = "black", main = "", xlab = xlabnames[i])
  hist(as.matrix(earnings_original)[!is.na(earnings[, i]), i],
       col = "white", border = "blue", add = TRUE)
  hist(as.matrix(earnings_original)[is.na(earnings[, i]), i],
       col = "white", border = "red", add = TRUE)
}
```



Each black histogram shows what the original data distribution looks like. The blue histograms represent the complete cases and the red ones represent the missing part.

After each of the following tasks, you need to implement the analysis you have in mind and report the results/estimates.

Step 2

Listwise deletion

This method is also called the complete cases method. It removes all observations from the dataset that have any missing values.

```
set.seed(1234)
# Listwise deletion
earnings.listwise <- na.omit(earnings)

# fit the regression
fit2 <- lm(earn ~ height + weight + male + education, data = earnings.listwise)
summary(fit2)
```

```
##
## Call:
## lm(formula = earn ~ height + weight + male + education, data = earnings.listwise)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -39378 -10658  -2118   5962 373596
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -37207.87   13392.10  -2.778  0.00554 **
## height         329.92     218.58   1.509  0.13143
## weight          11.34      20.00   0.567  0.57069
## male         10131.93    1607.33   6.304 3.94e-10 ***
## education     2302.31     213.20  10.799  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19750 on 1337 degrees of freedom
## Multiple R-squared:  0.1454, Adjusted R-squared:  0.1428
## F-statistic: 56.85 on 4 and 1337 DF,  p-value: < 2.2e-16
```

The summary of estimates and SE are presented in the summary table above.
So the estimated equation after listwise deletion method is

$$earn = -37207.86789 + 329.91661 height + 11.34224 weight + 10131.92932 male + 2302.30684 education$$

Step 3

Mean/mode imputation
For numerical variables, the mean imputation method in all missing values for a given variable with the mean of the observed values for that variable.
For categorical variables, the mode imputation method uses the value of variable's mode to impute the missing data.

```r
set.seed(1234)
earnings.mean_mode <- earnings # store the earnings to the earnings.mean_mode

# For each numerical variable which has missing values perform mean imputation
mean.imp <- function (a) {
  missing <- is.na(a)
  a.obs <- a[!missing]
  imputed <- a
  imputed[missing] <- mean(a.obs) # Output the imputed vector
  return(imputed)
}

earnings.mean_mode$height <- mean.imp(earnings.mean_mode$height) # height
earnings.mean_mode$weight <- mean.imp(earnings.mean_mode$weight) # weight
earnings.mean_mode$education <- mean.imp(earnings.mean_mode$education) # education


# For each categorical variable which has missing values perform mode imputation
mode <- function(x) {
  ta = table(x)
  tam = max(ta)
  if (all(ta == tam))
    mod = NA
  else
    mod = names(ta)[ta == tam]
  return(mod)
}
```

```
mode.imp <- function (a) {
  missing <- is.na(a)
  a.obs <- a[!missing]
  imputed <- a
  imputed[missing] <- mode(a.obs) # Output the imputed vector
  return (imputed)
}

earnings.mean_mode$male <- mode.imp(earnings.mean_mode$male) # male


# fit the regression
fit3 <- lm(earn ~ height + weight + male + education, data = earnings.mean_mode)
summary(fit3)
```

```
##
## Call:
## lm(formula = earn ~ height + weight + male + education, data = earnings.mean_mode)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -43341 -11319  -2647   6024 372085
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -62636.11   10807.97  -5.795 8.04e-09 ***
## height         654.53     176.00   3.719 0.000206 ***
## weight          13.37      17.21   0.777 0.437297
## male1         9006.76    1292.79   6.967 4.54e-12 ***
## education     2662.07     199.24  13.361  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20700 on 1782 degrees of freedom
## Multiple R-squared:  0.1666, Adjusted R-squared:  0.1647
## F-statistic: 89.07 on 4 and 1782 DF,  p-value: < 2.2e-16
```

The summary of estimates and SE are presented in the summary table above.
So the estimated equation after mean/mode imputation method is

$$earn = -62636.11 + 654.53height + 13.37weight + 9006.76male + 2662.07education$$

Step 4

Random imputation

The random imputation randomly picks observed value from the data and imputes the value to the missing part.

```
set.seed(1234)
earnings.random <- earnings # store the earnings to the earnings.random

random.imp <- function (a)
{
  missing <- is.na(a)
```

```r
  n.missing <- sum(missing)
  a.obs <- a[!missing]
  imputed <- a
  imputed[missing] <- sample (a.obs, n.missing, replace=TRUE)
  return (imputed)
}

earnings.random$height <- random.imp(earnings.random$height) # height
earnings.random$weight <- random.imp(earnings.random$weight) # weight
earnings.random$education <- random.imp(earnings.random$education) # education
earnings.random$male <- random.imp(earnings.random$male) # male

# fit the regression
fit4 <- lm(earn ~ height + weight + male + education, data = earnings.random)
summary(fit4)
```

```
##
## Call:
## lm(formula = earn ~ height + weight + male + education, data = earnings.random)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -41656 -11268  -2521   6098 372524
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -53671.668  10548.161  -5.088 3.99e-07 ***
## height         574.446    169.175   3.396   0.0007 ***
## weight           6.029     16.707   0.361   0.7182
## male          9280.720   1296.635   7.158 1.20e-12 ***
## education     2449.321    193.182  12.679  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20810 on 1782 degrees of freedom
## Multiple R-squared:  0.1578, Adjusted R-squared:  0.1559
## F-statistic: 83.47 on 4 and 1782 DF,  p-value: < 2.2e-16
```

The summary of estimates and SE are presented in the summary table above.
So the estimated equation after mean/mode imputation method is

$$earn = -53671.668 + 574.446 height + 6.029 weight + 9280.720 male + 2449.321 education$$

Step 5

LVCF (if applicable to your data)
Since the earnings data isn't a longitudinal data. This method doesn't seem to be applicable to the data.

Step 6

Hotdecking (nearest neighbor) with VIM package
The hotdecking method replaces missing values using other values found in the dataset. For each person with a missing value on variable Y, find another person who has all the same values (or close to the same values)

on observed variables X1, X2, X3…, and use that person's Y value.

```
set.seed(1234)

earnings.hotdecking <- earnings # store the earnings to the earnings.hotdecking

earnings.hotdecking <- hotdeck(earnings.hotdecking)[,1:5]

# fit the regression
fit6 <- lm(earn ~ height + weight + male + education, data = earnings.hotdecking)
summary(fit6)
```

```
##
## Call:
## lm(formula = earn ~ height + weight + male + education, data = earnings.hotdecking)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -41723 -11406  -2430   5953 372473
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -53600.65   10428.63  -5.140 3.05e-07 ***
## height         548.14     167.44   3.274  0.00108 **
## weight          14.19      16.35   0.868  0.38554
## male          9314.87    1285.04   7.249 6.24e-13 ***
## education     2476.79     192.46  12.869  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20760 on 1782 degrees of freedom
## Multiple R-squared:  0.1613, Adjusted R-squared:  0.1594
## F-statistic: 85.69 on 4 and 1782 DF,  p-value: < 2.2e-16
```

The summary of estimates and SE are presented in the summary table above.
So the estimated equation after mean/mode imputation method is

$$earn = -53600.65 + 548.14 height + 14.19 weight + 9314.87 male + 2476.79 education$$

Step 7

Regression imputation
Note you might have to use logistic or multinomial models, depending on what type of variable you impute values for.
Within the complete cases X_obs, build a model that predicts the values Y. And then use this model within the cases with missing data X_mis to predict (impute) Y.

```
set.seed(1234)

earnings.regression <- earnings # store the earnings to the earnings.hotdecking

# linear regression on numerical variables
# height
```

```r
earnings_height <- earnings.regression %>%
  select(height, earn)
Ry <- as.numeric(!is.na(earnings_height$height))
data.cc <- earnings_height[Ry == 1,]
data.dropped <- earnings_height[Ry == 0,]
reg <- lm(height ~ earn, data = data.frame(data.cc))
y.imp <- predict(reg, newdata = data.frame(data.dropped))
earnings_height$height[Ry == 0] <- y.imp

# weight
earnings_weight <- earnings.regression %>%
  select(weight, earn)
Ry <- as.numeric(!is.na(earnings_weight$weight))
data.cc <- earnings_weight[Ry == 1,]
data.dropped <- earnings_weight[Ry == 0,]
reg <- lm(weight ~ earn, data = data.frame(data.cc))
y.imp <- predict(reg, newdata = data.frame(data.dropped))
earnings_weight$weight[Ry == 0] <- y.imp

# education
earnings_education <- earnings.regression %>%
  select(education, earn)
Ry <- as.numeric(!is.na(earnings_education$education))
data.cc <- earnings_education[Ry == 1,]
data.dropped <- earnings_education[Ry == 0,]
reg <- lm(education ~ earn, data = data.frame(data.cc))
y.imp <- predict(reg, newdata = data.frame(data.dropped))
earnings_education$education[Ry == 0] <- y.imp

# logistic regression on binary vairable
# male
earnings_male <- earnings.regression %>%
  select(male, earn)
Ry <- as.numeric(!is.na(earnings_male$male))
data.cc <- earnings_male[Ry == 1,]
data.dropped <- earnings_male[Ry == 0,]

mylogit <- glm(male ~ earn, data = data.cc, family = "binomial")
y.imp <- predict(mylogit, newdata = data.dropped, type = "response")
earnings_male$male[Ry == 0] <- round(y.imp,0)


earnings.regression <- data.frame(cbind(height = earnings_height$height,
                            weight = earnings_weight$weight,
                            male = earnings_male$male,
                            education = earnings_education$education,
                            earn = earnings.regression$earn))

# fit the regression
fit7 <- lm(earn ~ height + weight + male + education, data = earnings.regression)
summary(fit7)
```

```
##
## Call:
```

```
## lm(formula = earn ~ height + weight + male + education, data = earnings.regression)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -44061 -11184  -2332   6149 371610
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -56558.50   10944.45  -5.168 2.63e-07 ***
## height         535.70     177.70   3.015  0.00261 **
## weight          13.99      17.00   0.823  0.41061
## male         11099.40    1305.50   8.502  < 2e-16 ***
## education     2724.20     195.33  13.947  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20340 on 1782 degrees of freedom
## Multiple R-squared:  0.1948, Adjusted R-squared:  0.193
## F-statistic: 107.8 on 4 and 1782 DF,  p-value: < 2.2e-16
```

The summary of estimates and SE are presented in the summary table above.
So the estimated equation after mean/mode imputation method is

$$earn = -56558.50 + 535.70 height + 13.99 weight + 11099.40 male + 2724.20 education$$

Step 8

Regression imputation with noise on all variables (numerical, dichotomous and multinomial). This method is basically like the method in step 7 but also add noises when predicting the missing values.

```
set.seed(1234)

earnings.regression_with_noise <- earnings # store the earnings to the earnings.hotdecking

# linear regression on numerical variables
# height
earnings_height <- earnings.regression_with_noise %>%
  select(height, earn)
Ry <- as.numeric(!is.na(earnings_height$height))
data.cc <- earnings_height[Ry == 1,]
data.dropped <- earnings_height[Ry == 0,]
reg <- lm(height ~ earn, data = data.frame(data.cc))
y.imp <- predict(reg, newdata = data.frame(data.dropped))
y.imp <- y.imp + rnorm(length(y.imp), 0, summary(reg)$sigma)# noise
earnings_height$height[Ry == 0] <- y.imp

# weight
earnings_weight <- earnings.regression_with_noise %>%
  select(weight, earn)
Ry <- as.numeric(!is.na(earnings_weight$weight))
data.cc <- earnings_weight[Ry == 1,]
data.dropped <- earnings_weight[Ry == 0,]
reg <- lm(weight ~ earn, data = data.frame(data.cc))
```

```r
y.imp <- predict(reg, newdata = data.frame(data.dropped))
y.imp <- y.imp + rnorm(length(y.imp), 0, summary(reg)$sigma) # noise
earnings_weight$weight[Ry == 0] <- y.imp

# education
earnings_education <- earnings.regression_with_noise %>%
  select(education, earn)
Ry <- as.numeric(!is.na(earnings_education$education))
data.cc <- earnings_education[Ry == 1,]
data.dropped <- earnings_education[Ry == 0,]
reg <- lm(education ~ earn, data = data.frame(data.cc))
y.imp <- predict(reg, newdata = data.frame(data.dropped))
y.imp <- y.imp + rnorm(length(y.imp), 0, summary(reg)$sigma) # noise
earnings_education$education[Ry == 0] <- y.imp

# logistic regression on binary variable
# male
earnings_male <- earnings.regression_with_noise %>%
  select(male, earn)
Ry <- as.numeric(!is.na(earnings_male$male))
data.cc <- earnings_male[Ry == 1,]
data.dropped <- earnings_male[Ry == 0,]

mylogit <- glm(male ~ earn, data = data.cc, family = "binomial")
y.imp <- predict(mylogit, newdata = data.dropped, type = "response")
earnings_male$male[Ry == 0] <- rbinom(sum(Ry == 0), 1, y.imp)


earnings.regression_with_noise <- data.frame(cbind(height = earnings_height$height,
                            weight = earnings_weight$weight,
                            male = earnings_male$male,
                            education = earnings_education$education,
                            earn = earnings.regression_with_noise$earn))

# fit the regression
fit8 <- lm(earn ~ height + weight + male + education, data = earnings.regression_with_noise)
summary(fit8)
```

```
##
## Call:
## lm(formula = earn ~ height + weight + male + education, data = earnings.regression_with_noise)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -42424 -11333  -2395   6193 372188
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -55292.61   10593.77  -5.219 2.01e-07 ***
## height         526.32     169.22   3.110   0.0019 **
## weight          24.95      16.25   1.535   0.1249
## male          9675.52    1296.26   7.464 1.30e-13 ***
## education     2576.59     191.23  13.474  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20560 on 1782 degrees of freedom
## Multiple R-squared:  0.178,  Adjusted R-squared:  0.1761
## F-statistic: 96.44 on 4 and 1782 DF,  p-value: < 2.2e-16
```

The summary of estimates and SE are presented in the summary table above.
So the estimated equation after mean/mode imputation method is

$$earn = -55292.61 + 526.32height + 24.95weight + 9675.52male + 2576.59education$$

Multiple imputation with either mice OR mi package

Step 9

Load your data into the package. Obtain summary, and graphs of the data and missing patterns.
This is the summary of the data.

```
# summary of the data
summary(earnings)
```

```
##      height          weight           male          education
##  Min.   :57.00   Min.   : 80.0   Min.   :0.0000   Min.   : 2.00
##  1st Qu.:64.00   1st Qu.:130.0   1st Qu.:0.0000   1st Qu.:12.00
##  Median :66.00   Median :150.0   Median :0.0000   Median :12.00
##  Mean   :66.53   Mean   :155.2   Mean   :0.3608   Mean   :13.23
##  3rd Qu.:69.00   3rd Qu.:175.0   3rd Qu.:1.0000   3rd Qu.:15.00
##  Max.   :82.00   Max.   :342.0   Max.   :1.0000   Max.   :18.00
##  NA's   :104     NA's   :107     NA's   :110      NA's   :124
##       earn
##  Min.   :     0
##  1st Qu.:  6000
##  Median : 16000
##  Mean   : 21248
##  3rd Qu.: 27000
##  Max.   :400000
##
```

Using the flux() function to obtain more detailed summary statistics per variable. The summary table is shown below.

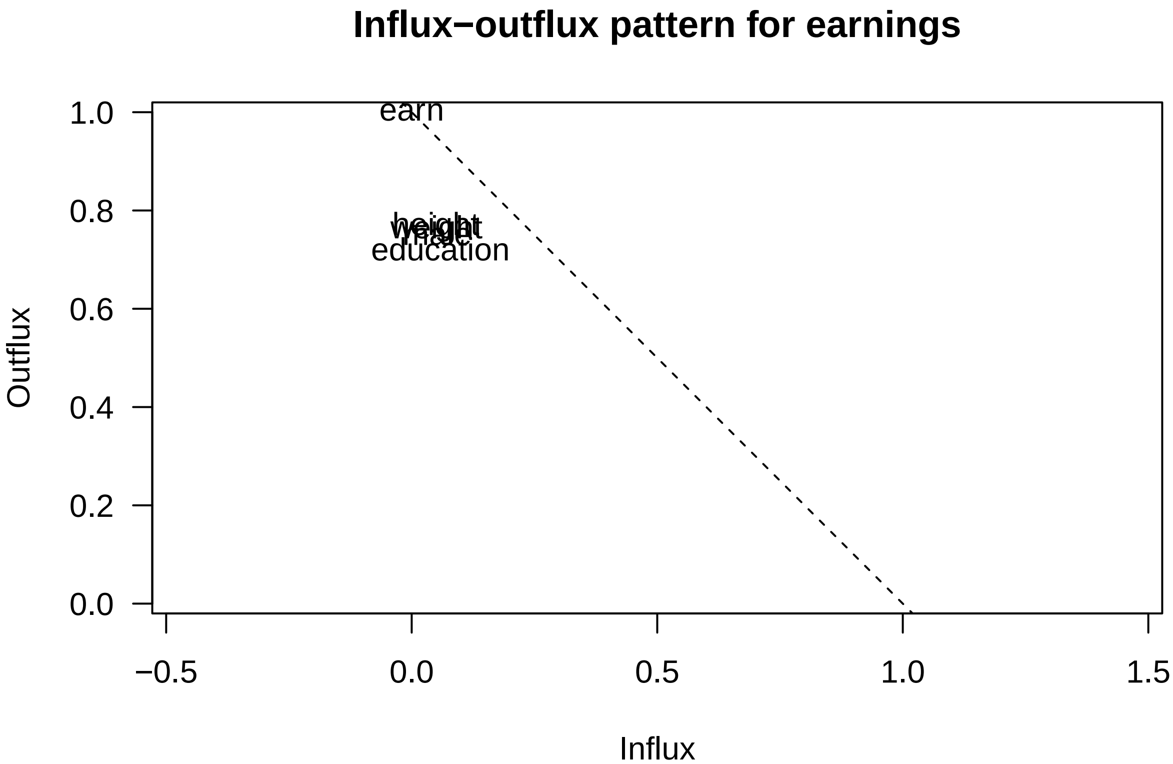
```
# More detailed summary statistics per variable
fluxsummary <- flux(earnings)
knitr::kable(fluxsummary)
```

	pobs	influx	outflux	ainb	aout	fico
height	0.9418019	0.0489988	0.7662921	1	0.0506536	0.2026144
weight	0.9401231	0.0504122	0.7595506	1	0.0502976	0.2011905
male	0.9384443	0.0518257	0.7528090	1	0.0499404	0.1997615
education	0.9306100	0.0584217	0.7213483	1	0.0482562	0.1930247

	pobs	influx	outflux	ainb	aout	fico
earn	1.0000000	0.0000000	1.0000000	0	0.0622552	0.2490207

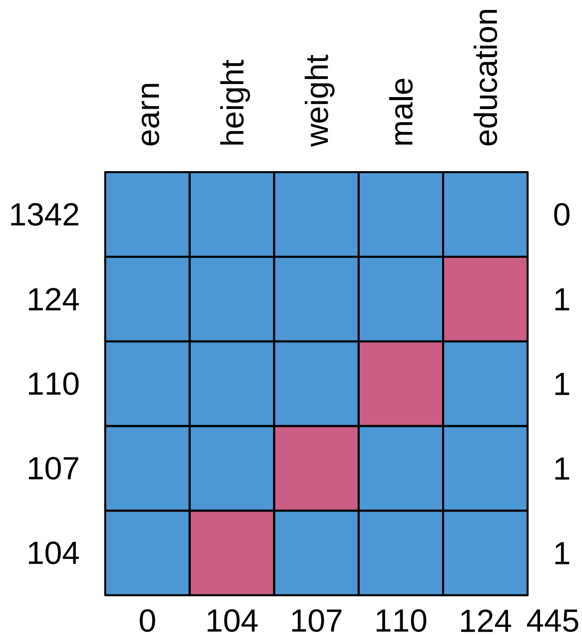
The histograms of the 4 variables with missing data has already been presented in Step 1. So let's take a look at other information graph of the data.

```
# graphs of the data
fluxplot(earnings)
```



Then check the missing patterns

```
earnings.mice_mi <- missing_data.frame(earnings) # store the earnings as the missing data frame

md.pattern(earnings.mice_mi, rotate.names = T) # check the pattern
```




```
##      earn height weight male education
## 1342    1      1      1    1         1   0
## 124     1      1      1    1         0   1
## 110     1      1      1    0         1   1
## 107     1      1      0    1         1   1
## 104     1      0      1    1         1   1
##         0    104    107  110       124 445
```

```
# look at the patterns numerically
tabulate(earnings.mice_mi@patterns)
```

```
## [1] 1342  124  104  110  107
```

```
levels(earnings.mice_mi@patterns)
```

```
## [1] "nothing"   "education" "height"    "male"      "weight"
```

So there are five missingness patterns. 1342 cases had "nothing" missingness pattern, 124 cases had "education" missingness pattern, 104 cases had "height" missingness pattern, 110 cases had "male" missingness pattern, 107 cases had "weight" missingness pattern.

Step 10

Check your data types and methods and make changes if necessary.
The data types of "height", "weight", "education",and "earn" are numerical and "male" is a binary variable.

```
show(earnings.mice_mi)
```

```
## Object of class missing_data.frame with 1787 observations on 5 variables
##
## There are 5 missing data patterns
##
## Append '@patterns' to this missing_data.frame to access the corresponding pattern for every observati
##
##                  type missing method  model
```

```
## height    continuous     104    ppd linear
## weight    continuous     107    ppd linear
## male          binary     110    ppd  logit
## education continuous     124    ppd linear
## earn      continuous       0   <NA>   <NA>
##
##              family     link transformation
## height    gaussian identity    standardize
## weight    gaussian identity    standardize
## male      binomial    logit           <NA>
## education gaussian identity    standardize
## earn          <NA>     <NA>    standardize
```

According to the table, there is no need to make changes.

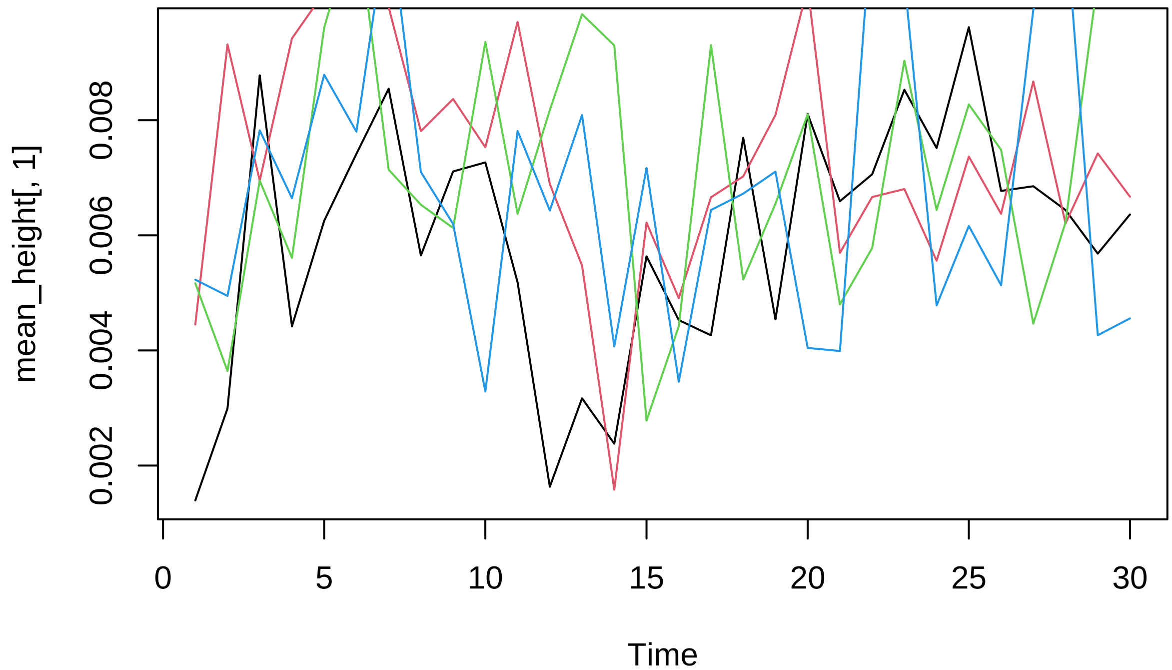
Step 11

Run the mi/mice command and check convergence by traceplots.
First, run the mi command.

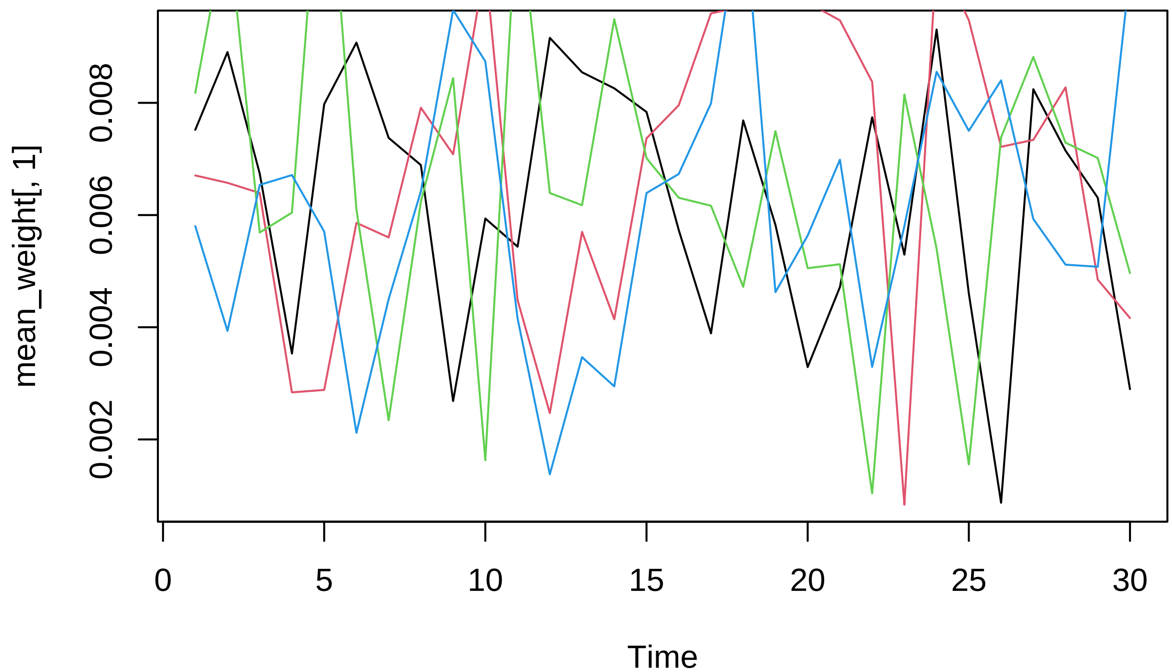
```
# run the mi command
imp.earnings <- mi(earnings.mice_mi, seed = 1, parallel = F)
```

Then, check the convergence.

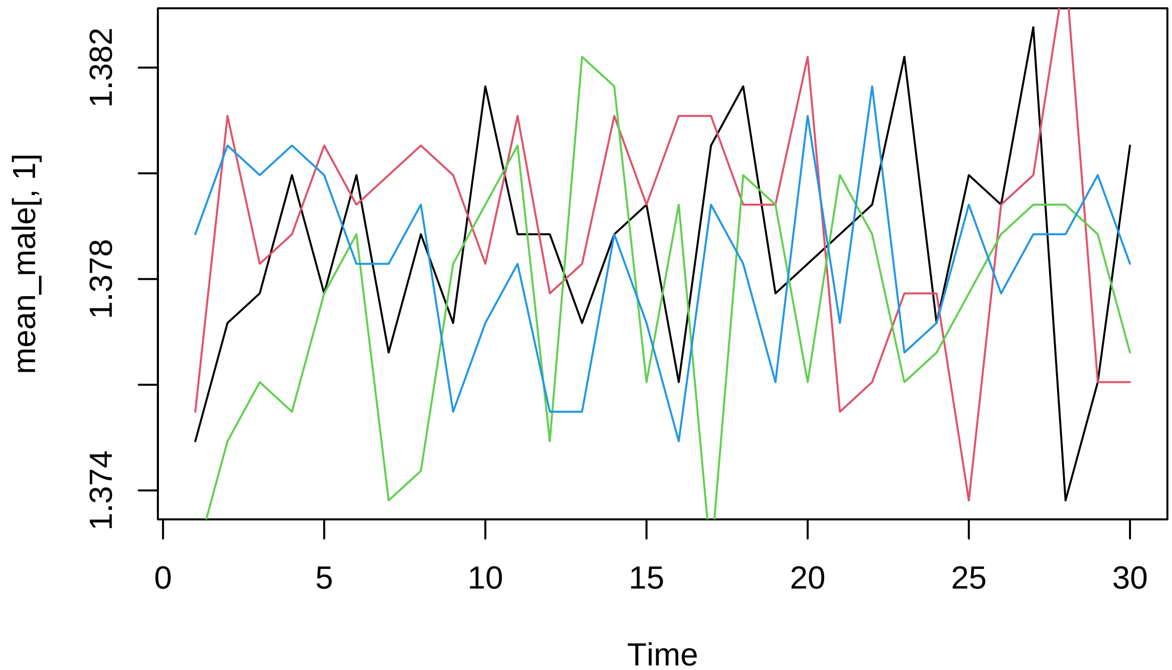
```
converged <- mi2BUGS(imp.earnings)

mean_height = converged[, , 1]
mean_weight = converged[, , 2]
mean_male = converged[, , 3]
mean_education = converged[, , 4]

ts.plot(mean_height[,1], col=1)
lines(mean_height[,2], col= 2)
lines(mean_height[,3], col= 3)
lines(mean_height[,4], col= 4)
```



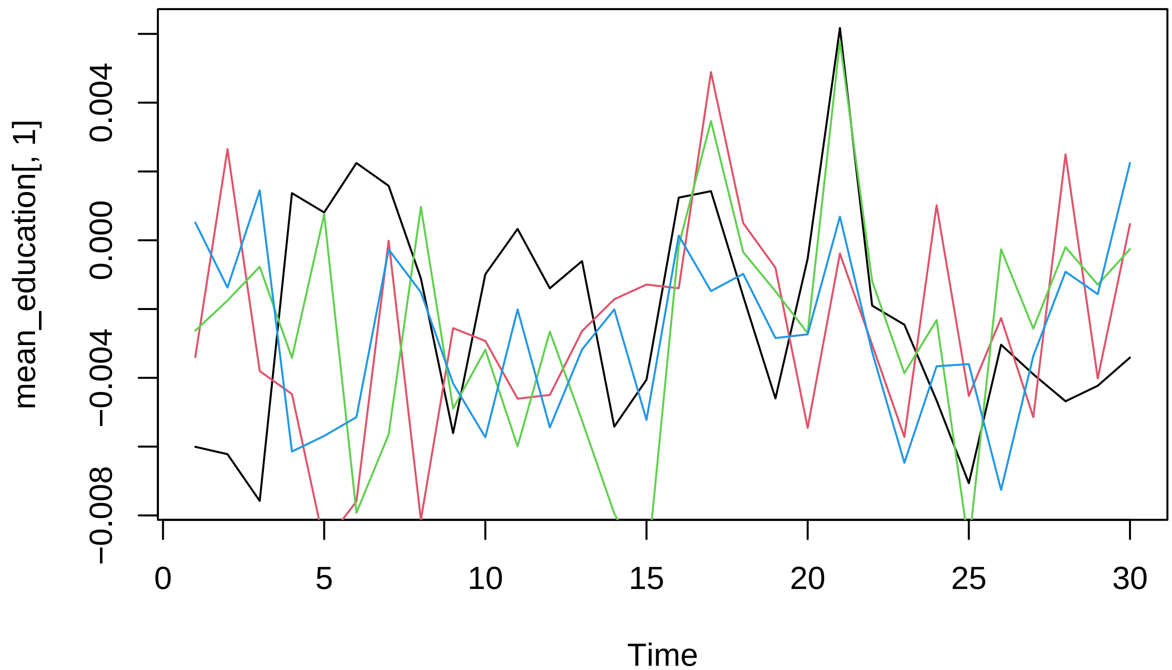

```
ts.plot(mean_weight[,1], col=1)
lines(mean_weight[,2], col= 2)
lines(mean_weight[,3], col= 3)
lines(mean_weight[,4], col= 4)
```




```
ts.plot(mean_male[,1], col=1)
lines(mean_male[,2], col= 2)
lines(mean_male[,3], col= 3)
lines(mean_male[,4], col= 4)
```




```
ts.plot(mean_education[,1], col=1)
lines(mean_education[,2], col= 2)
lines(mean_education[,3], col= 3)
lines(mean_education[,4], col= 4)
```



Step 12

Check r-hats

The r-hats are shown in the table below.

```r
r_hats <- Rhats(imp.earnings)
r_hats <- as.data.frame(r_hats)
knitr::kable(r_hats)
```

	r_hats
mean_height	1.0221134
mean_weight	0.9916843
mean_male	1.0218507
mean_education	0.9856015
sd_height	1.0231633
sd_weight	0.9889921
sd_male	1.0220418
sd_education	0.9923803

Step 13

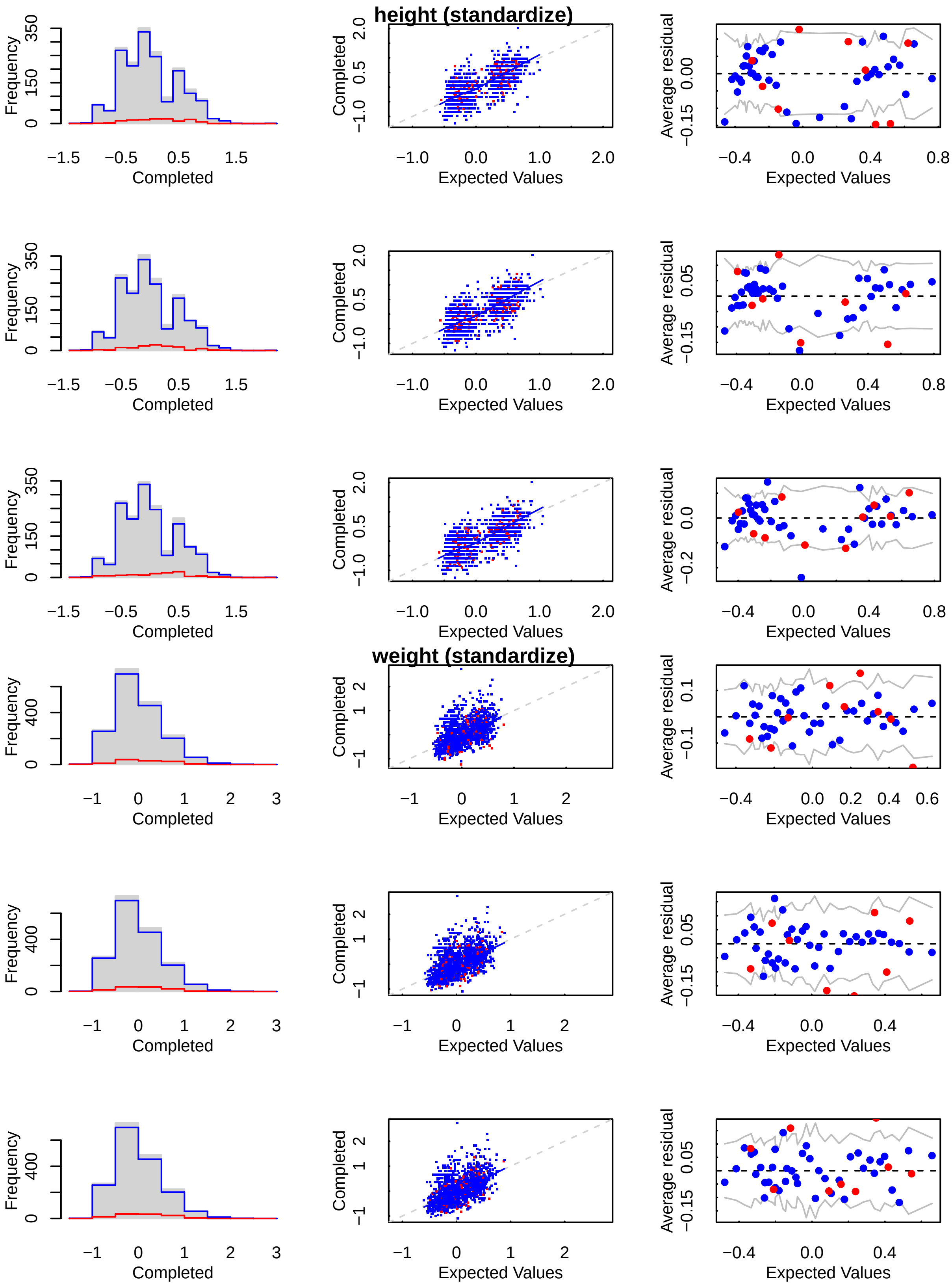
Increase number of imputations if necessary
In this step, I change the iteration times to 50, while the previous defaulting is 30.

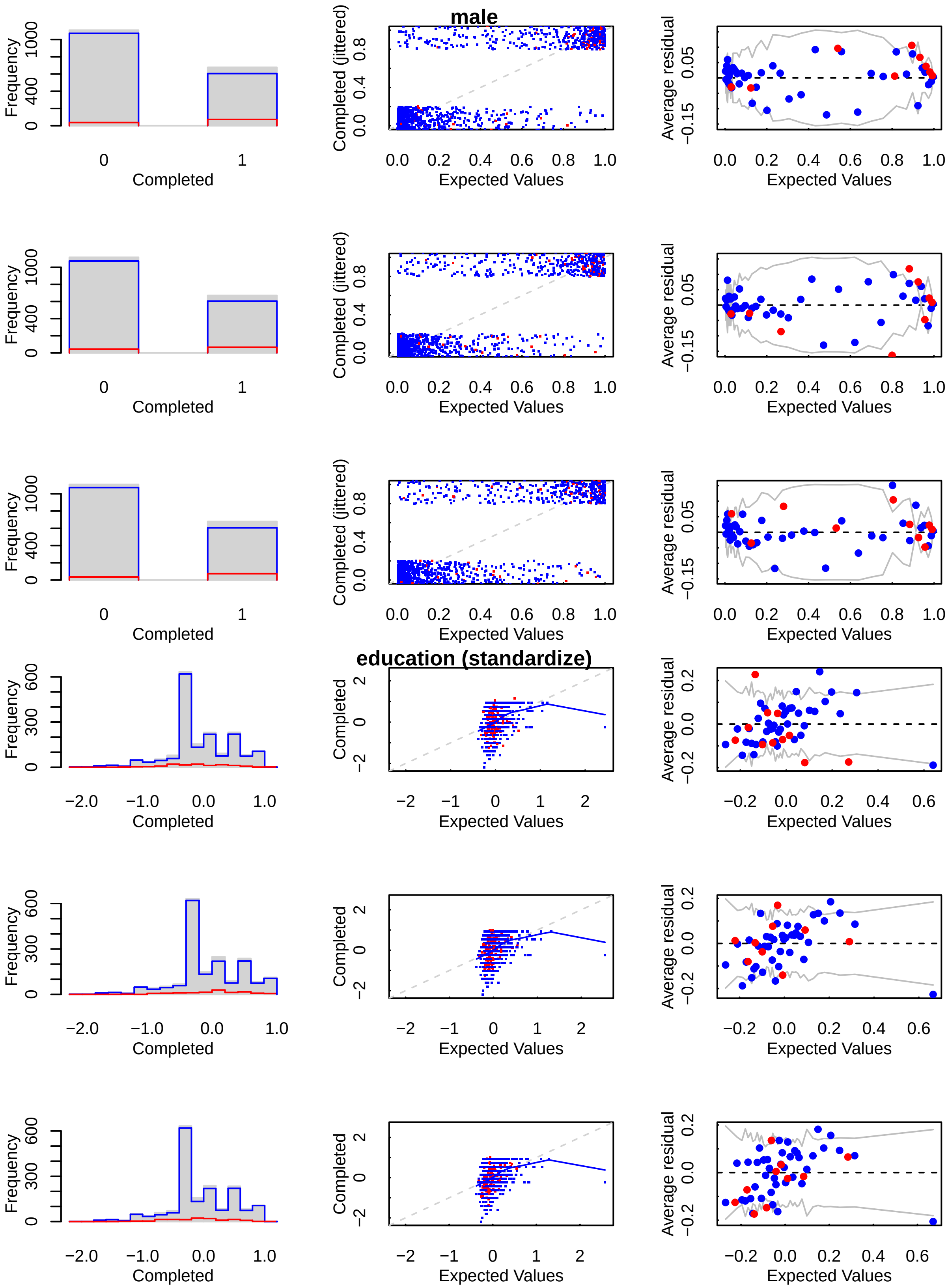
```r
imp.earnings <- mi(earnings.mice_mi, n.iter = 50, seed = 1, parallel = F)
```

Step 14

Plot some diagnostics

```r
plot(imp.earnings)
```


height (standardize)
weight (standardize)
Frequency
Completed
Expected Values
Average residual
0
150
350
400
−1.5
−0.5
0.5
1.5
−1.0
0.0
1.0
2.0
−0.4
0.0
0.2
0.4
0.6
0.8
−0.15
0.00
0.05
−0.2
0.1
−0.1
−1
1
2
3



From the plot, we can see the imputation for education isn't ideal(from the picture of education in the middle) and the distribution of "height", "weight" and "education" are still a bit different from the observed data. So we need step 15.

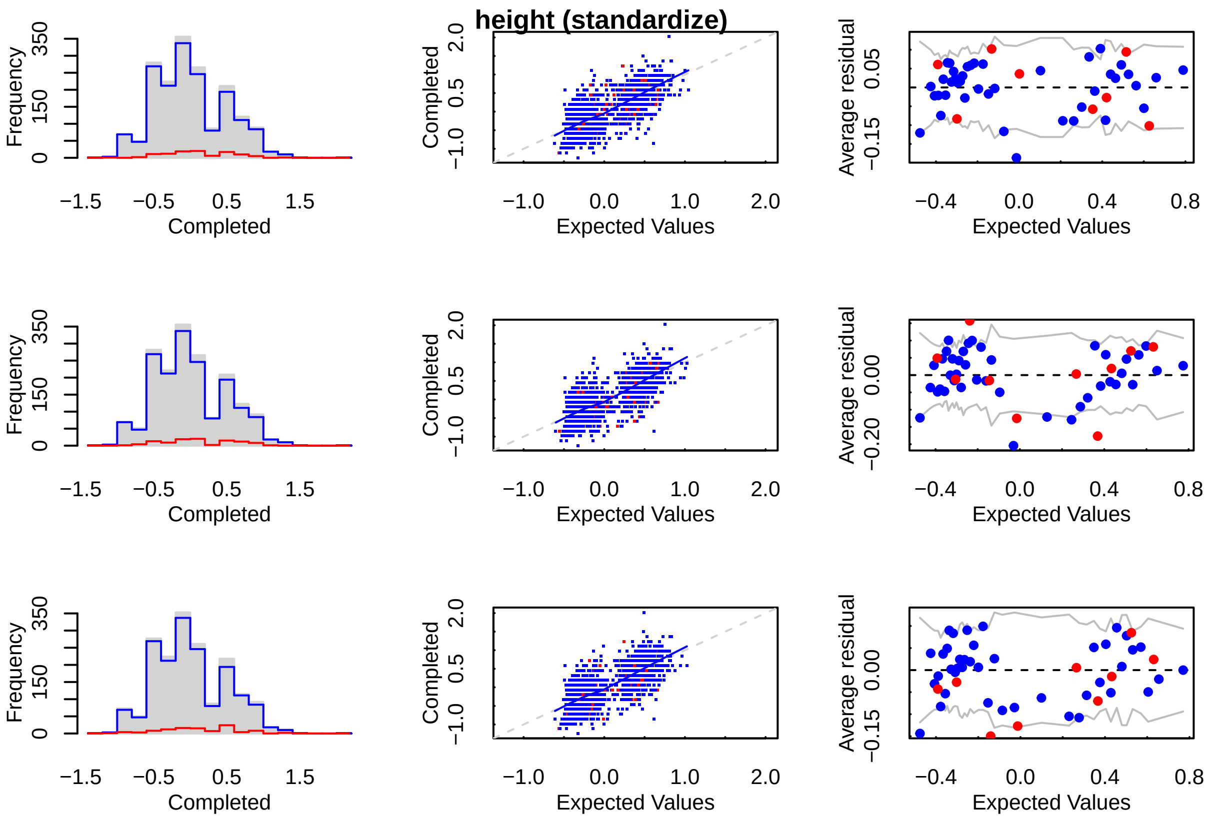
Step 15

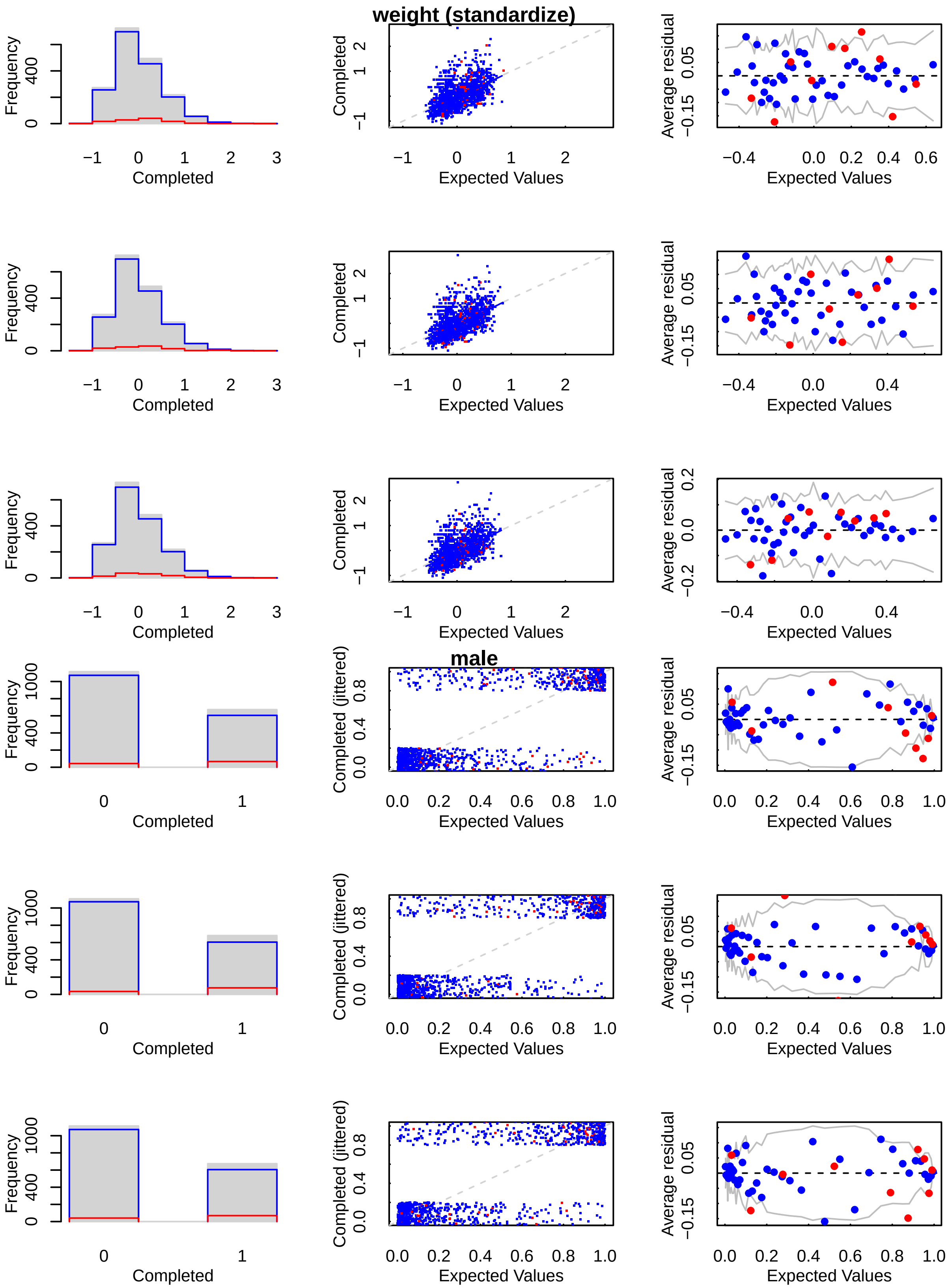
Change imputation models if necessary, and/or number of chains.
Change the "education" type to "positive" and change the imputation method for "height", "weight", and "education" to "pmm".

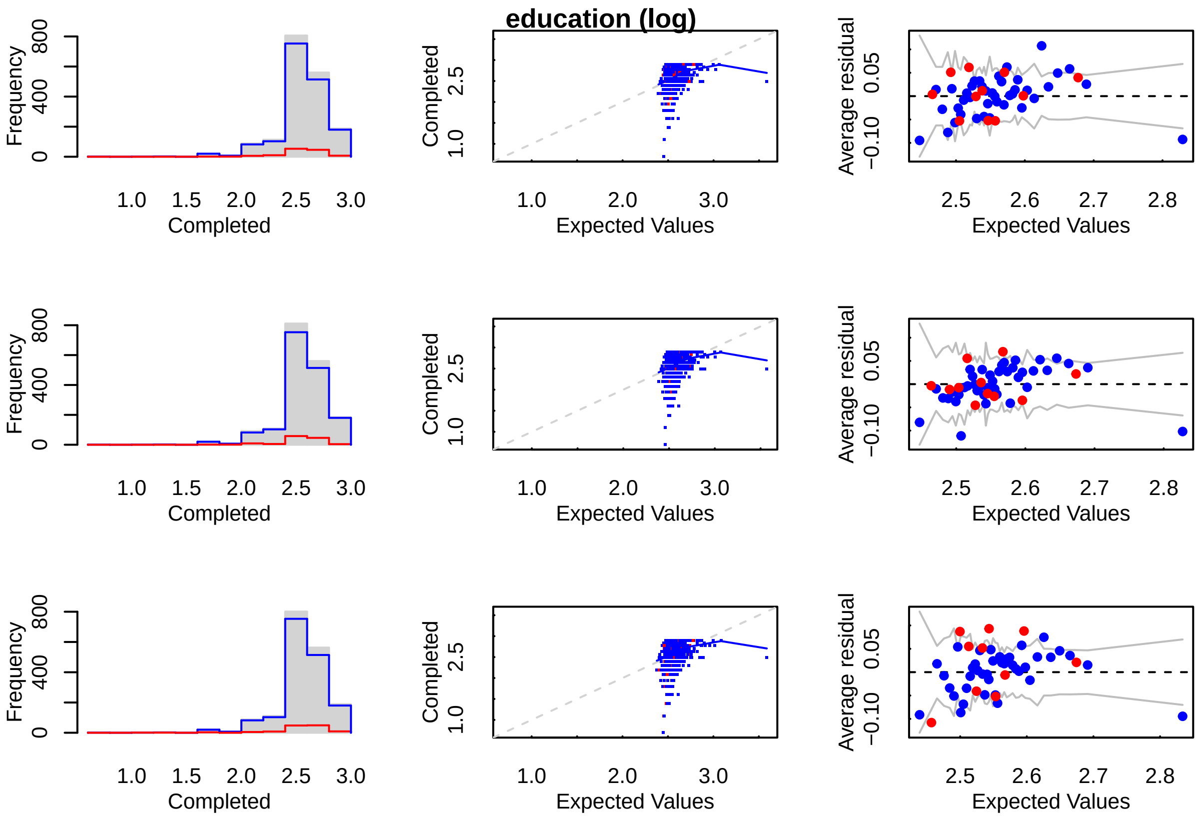
```
set.seed(1234)

earnings.mice_mi <- change(earnings.mice_mi, y = "education",
                           what = "type", to = "pos")
earnings.mice_mi <- change(earnings.mice_mi , y = c("height", "weight", "education"),
                           what = "imputation_method", to = "pmm")
imp.earnings <- mi(earnings.mice_mi, n.chains = 5, n.iter = 50, seed = 1, parallel = F)
```

Now let's plot the diagnostics again.

```
plot(imp.earnings)
```




weight (standardize)
male
Frequency
Completed
Completed (jittered)
Expected Values
Average residual



The imputation for education is much better now. And the distributions of "height", "weight" and "education" are more similar(similar peaks) to the observed data distribution.

Step 16

Run pooled analysis

```
set.seed(1234)
fit9 = mi::pool(earn ~ height + weight + male + education, data=imp.earnings)
display(fit9)
```

```
## bayesglm(formula = earn ~ height + weight + male + education,
##     data = imp.earnings)
##             coef.est  coef.se
## (Intercept) -44261.84  12193.59
## height         375.48    199.12
## weight          11.37     18.33
## male1        11246.79   1467.80
## education     2609.74    193.36
## n = 1782, k = 5
## residual deviance = 7.47948e+11, null deviance = 915918830442.7 (difference = 167970863790.3)
## overdispersion parameter = 419723887.0
## residual sd is sqrt(overdispersion) = 20487.16
```

The summary of estimates and SE are presented in the summary table above.
So the estimated equation after using mi is

$$earn = -44261.84 + 375.48height + 11.37weight + 11246.79male + 2609.74education$$

Combined summary of results

Step 17

Prepare a table with results from all imputation methods

```
coefs <- matrix(NA, nrow = 7, ncol = 5)
ses <- matrix(NA, nrow = 7, ncol = 5)

colnames(coefs) <- c("Intercept Est", "height Est",
                     "weight Est", "male Est", "education Est")
rownames(coefs) <- c("listwise", "mean/mode", "random", "hotdecking",
                     "regression", "reg with noise", "mi")
colnames(ses) <- c("Intercept SE", "height SE",
                   "weight SE", "male SE", "education SE")
rownames(ses) <- c("listwise", "mean/mode", "random", "hotdecking",
                   "regression", "reg with noise", "mi")

coefs[1, ] <- summary(fit2)$coefficients[1:5,1]
ses[1, ] <- summary(fit2)$coefficients[1:5,2]

coefs[2, ] <- summary(fit3)$coefficients[1:5,1]
ses[2, ] <- summary(fit3)$coefficients[1:5,2]

coefs[3, ] <- summary(fit4)$coefficients[1:5,1]
ses[3, ] <- summary(fit4)$coefficients[1:5,2]

coefs[4, ] <- summary(fit6)$coefficients[1:5,1]
ses[4, ] <- summary(fit6)$coefficients[1:5,2]

coefs[5, ] <- summary(fit7)$coefficients[1:5,1]
ses[5, ] <- summary(fit7)$coefficients[1:5,2]

coefs[6, ] <- summary(fit8)$coefficients[1:5,1]
ses[6, ] <- summary(fit8)$coefficients[1:5,2]

coefs[7, ] <- summary(fit9)$coefficients[1:5,1]
ses[7, ] <- summary(fit9)$coefficients[1:5,2]

one_final_table <- t(cbind(coefs, ses))

knitr::kable(one_final_table)
```

	listwise	mean/mode	random	hotdecking	regression	reg with noise	mi
Intercept Est	-37207.86789	-62636.10943	-53671.667969	-53600.64879	-56558.50331	-55292.60741	-44261.83998
height Est	329.91661	654.52593	574.445633	548.14199	535.70467	526.32004	375.48485
weight Est	11.34224	13.36888	6.029178	14.19090	13.98991	24.94667	11.36636
male Est	10131.92932	9006.76003	9280.720489	9314.87050	11099.39563	9675.52026	11246.78603
education Est	2302.30684	2662.07337	2449.320826	2476.78837	2724.19935	2576.58883	2609.74234
Intercept SE	13392.10464	10807.96595	10548.160746	10428.62942	10944.44822	10593.77086	12193.58632

	listwise	mean/mode	random	hotdecking	regression	reg with noise	mi
height SE	218.57585	176.00076	169.175294	167.44377	177.69546	169.22383	199.11843
weight SE	19.99778	17.20709	16.707464	16.34992	16.99828	16.24913	18.32523
male SE	1607.32855	1292.78796	1296.635011	1285.03990	1305.50085	1296.26112	1467.80323
education SE	213.20408	199.24058	193.182041	192.46437	195.32976	191.22747	193.36082

The table above shows the coefficient and SE result for each methods used in this project.
Each row corresponds to the estimates of the parameters along with SE of each imputation method.

Discussion

Step 18

Discuss and compare to original data in terms of average percent change in coefficients and SE

```r
# original data fit the regression
fit1 <- lm(earn ~ height + weight + male + education, data = earnings_original)
summaryfit1 <- summary(fit1)
summaryfit1
```

```
##
## Call:
## lm(formula = earn ~ height + weight + male + education, data = earnings_original)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -41748 -11353  -2326   6164 372736
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -39967.17   11773.65  -3.395 0.000702 ***
## height         305.40     192.79   1.584 0.113344
## weight          16.32      17.02   0.959 0.337698
## male         11375.48    1435.37   7.925 3.98e-15 ***
## education     2570.64     190.98  13.460  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20510 on 1782 degrees of freedom
## Multiple R-squared:  0.1816, Adjusted R-squared:  0.1798
## F-statistic: 98.88 on 4 and 1782 DF,  p-value: < 2.2e-16
```

```r
ave_change.coef <- rep(NA, 7)
ave_change.se <- rep(NA, 7)

for (i in 1:7){
  # average percent change in coefficients
  ave_change.coef[i] <- mean(abs(summaryfit1$coefficients[1:5,1] - coefs[i, ])
                             /abs(summaryfit1$coefficients[1:5,1]))

  # average percent change in SE
```

```
  ave_change.se[i] <- mean(abs(summaryfit1$coefficients[1:5,2] - ses[i, ])
                          /abs(summaryfit1$coefficients[1:5,2]))
}

ave_change <- rbind(ave_change.coef,ave_change.se)
colnames(ave_change) <- c("listwise", "mean/mode", "random", "hotdecking",
                          "regression", "reg with noise", "mi")
rownames(ave_change) <- c("coefficients ave change","se ave change")

knitr::kable(ave_change)
```

	listwise	mean/mode	random	hotdecking	regression	reg with noise	mi
coefficients ave change	0.1335791	0.4269657	0.4171352	0.2967784	0.2791695	0.3574850	0.1333729
se ave change	0.1365295	0.0645911	0.0705646	0.0794581	0.0525895	0.0731321	0.0361047

So the estimated equation of the original data is

$$earn = -39967.17 + 305.40 height + 16.32 weight + 11375.48 male + 2570.64 education$$

The table shows the average percent change in coefficients and SE and also shows that the method using mi package gives the smallest change both in coefficients(13.33729%) and SE(3.61047%) which may indicates that this is the best methods for imputing missing data for this data set.