Incomplete Data Analysis

Assignment 1

Josephine Li s2346729

Question 1

Sub-Question (a)

As we suppose that ALQ is MCAR, the probability of missing value has no relation with if the values themselves are observed or other variables. Therefore, the probability of missing for AQL is unrelated to if the answer is "Yes" or "No", which means the probability of missing for those with ALQ = No is the same as for those with ALQ = Yes.

The answer for this question is **(ii) 0.3**.

Sub-Question(b)

In this part, as ALQ is MAR given by gender, the probability of missing for ALQ depends on gender and is further unrelated to specific missing values, which is if ALQ = Yes or ALQ = No.

The answer for this question is (ii).

Sub-Question(c)

In this survey, the gender value for participants are complete, and the gender for this survey only consists "male" and "female", therefore, a participant for this survey can only be a man or a woman. Suppose that:

$$A : the\ value\ for\ AQL\ is\ missing$$

$$B : \begin{cases} B_1, & male \\ B_2, & female \end{cases}$$

According to the total probability theorem $P(A) = P(A|B_1) \times P(B_1) + p(A|B_2) \times P(B_2)$.

However, we only have the probability of $P(A|B_1)$, which is 0.1, therefore, we cannot calculate the probability of ALQ being missing for women, which is $P(A|B_2)$.

The answer for this question is (iii).

Question 2

A dataset consists of 100 subjects and 10 variables will have $100 \times 10 = 1000$ recordings (include NA). To simplify description, I will set those missing value's indexes larger than those being observed and fill observed value by 1.

- **Case 1:** Largest Possible Subsample under a Complete Data Analysis

  In this case, to get a largest subsample, we can assume that subject1 to subject90's 10 variavles' recordings are complete and subject91 to subject100's 10 variables are missing. The left part in the below table is a example of the total dataset under this case, the right part of the below table is the corresponding largest possible subsample.

	Sub1	**...**	**Sub90**	**Sub91**	**...**	**Sub100**		**Sub1**	**...**	**Sub90**
Var1	1	...	1	NA	...	NA	**Var1**	1	...	1
Var2	1	...	1	NA	...	NA	**Var2**	1	...	1
...							**...**			
Var10	1	...	1	NA	...	NA	**Var10**	1	...	1

  To generate to a more common situation, if the missing data are all concentrate on same subjects (those subjects' variables recording are all missing), we can get the largest subsample under a complete data analysis.

- **Case 2:** Smallest Possible Subsample under a Complete Data Analysis

  In this case, to get a smallest subsample, we can assume that subject1 to subject10's variable1's recordings are missing, ubject11 to subject20's variable2's recordings are missing..., and so on. The table below is the total dataset in this example, and the smallest subsample under a complete data analysis is empty.

	Sub1-10	**Sub11-20**	**...**	**Sub81-90**	**Sub91-100**
Var1	**NA**	1	...	1	1
Var2	1	**NA**	...	1	1
...	...	...	**NA**	...	...
Var9	1	1	...	**NA**	1
Var10	1	1		1	**NA**

  To generate to a more common situation, if every subjects have at least 1 missing data and those missing data can satisfy the condition that every variables have 10% missing data, then we can obtain the smallest subsample, which is a empty set, under a complete data analysis.

Question 3

Sub-Question (a)

1. Simulate a dataset of size 500 on (Y_1, Y_2)

```r
# simulate random seed
set.seed(1)
z1 <- rnorm(500)
z2 <- rnorm(500)
z3 <- rnorm(500)

# initialize y1, y2
y1 <- z1 + 1
y2_original <- 5 + 2*z1 + z2
```

2. Simulate the corresponding observed dataset $(a = 2, b = 0)$

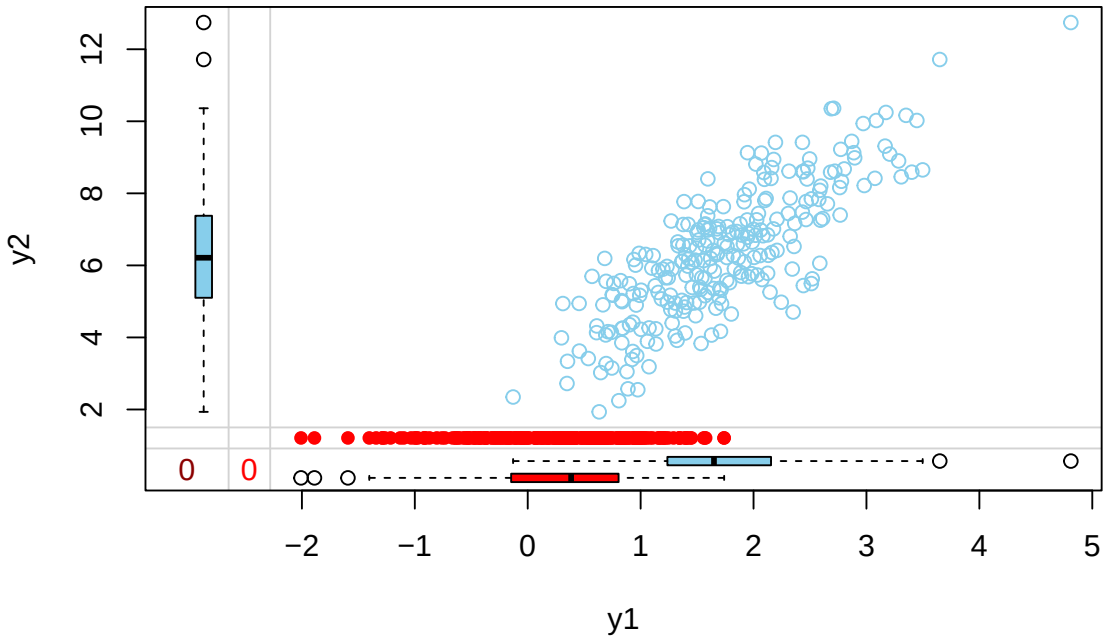
```r
# simulate corresponding observed data (a = 2, b = 0)
a <- 2
b <- 0
# eveluate if y2 is missing
missing_evaluate <- (a*(y1-1) + b*(y2_original-5) + z3)
y2_miss_index <- which(missing_evaluate < 0)

# get 'real' y2(after missing)
y2 <- y2_original
for (i in y2_miss_index){
  y2[i] <- NA}
```

3. Display the marginal distribution of Y_2 and evaluate the mechanism.

   - Comparing Y_1 and Y_2

```r
# display the margnial distribution(y1,y2)
require(VIM)
Y <- data.frame(y1,y2)
marginplot(Y)
```

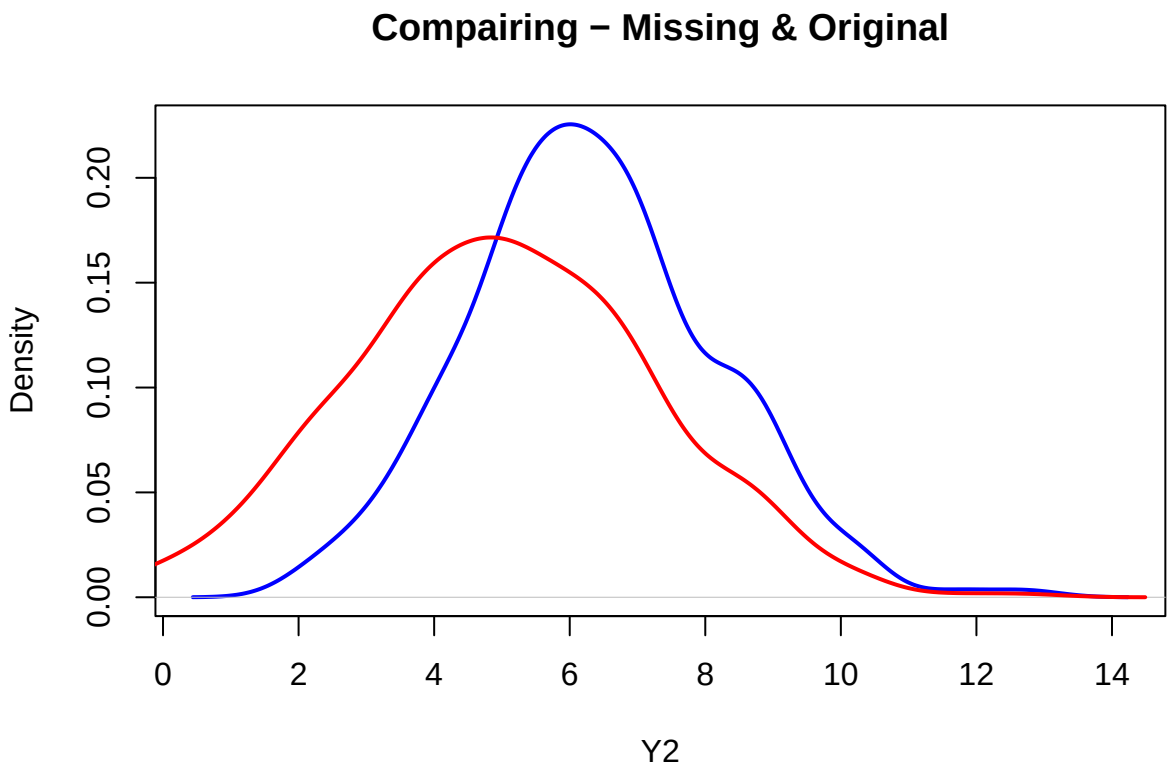


The blue point in the largest area correspond to observations which both Y_1 and Y_2 are observed. The red dots below correspond to records which Y_1 is observed while Y_2 is missing. The red boxplot below represents the marginal distribution of Y_1 observation which the corresponding Y_2 are missing. The blue boxplot below represents the marginal distribution of Y_1 which the corresponding Y_2 are observed. The blue boxplot on the left represents the marginal distribution of Y_2.

We can see from 2 boxplots below that the marginal distributions of Y_1 have an obvious difference when Y_2 is observed or missing. The mechanism is not MCAR, as if it is, the 2 distributions are expected to be identity.

- Comparing original Y_2 (before missing) and 'real' Y_2 (after missing)

```
# y2_obs is a vector only contain y2 observed values
y2_obs <- c()
for (i in y2){
  if (!(is.na(i))){
    y2_obs <- c(y2_obs,i)
  }
}
# plot marginal distributions
plot(density(y2_obs), lwd = 2, col = "blue", xlab = "Y2",
     main = "Compairing - Missing & Original")
lines(density(y2_original), lwd = 2, col = "red")
```



We can see that the observed data of Y_2 (representing by the red curve) and the 'complete' data of Y_2 (representing by the blue curve) have similar distribution.

In real situation, as we cannot observed the missing value, we cannot absolutely say that the mechanism is MNAR or MAR. In this graph, as 2 Y_2 have similar shape in distributions. Therefore, I hold the opinion that it has a higher probability to be MAR than MNAR.

However, as those 'missing values' are generated by ourselves by using $a \times (Y_1 - 1) + b \times (Y_2 - 5) + z_3$ where $a = 2, b = 0$, we will know that the missing of Y_2 have no relation with itself. So, by this information, we can say that the mechanism is **MAR**.

Sub-Question (b)

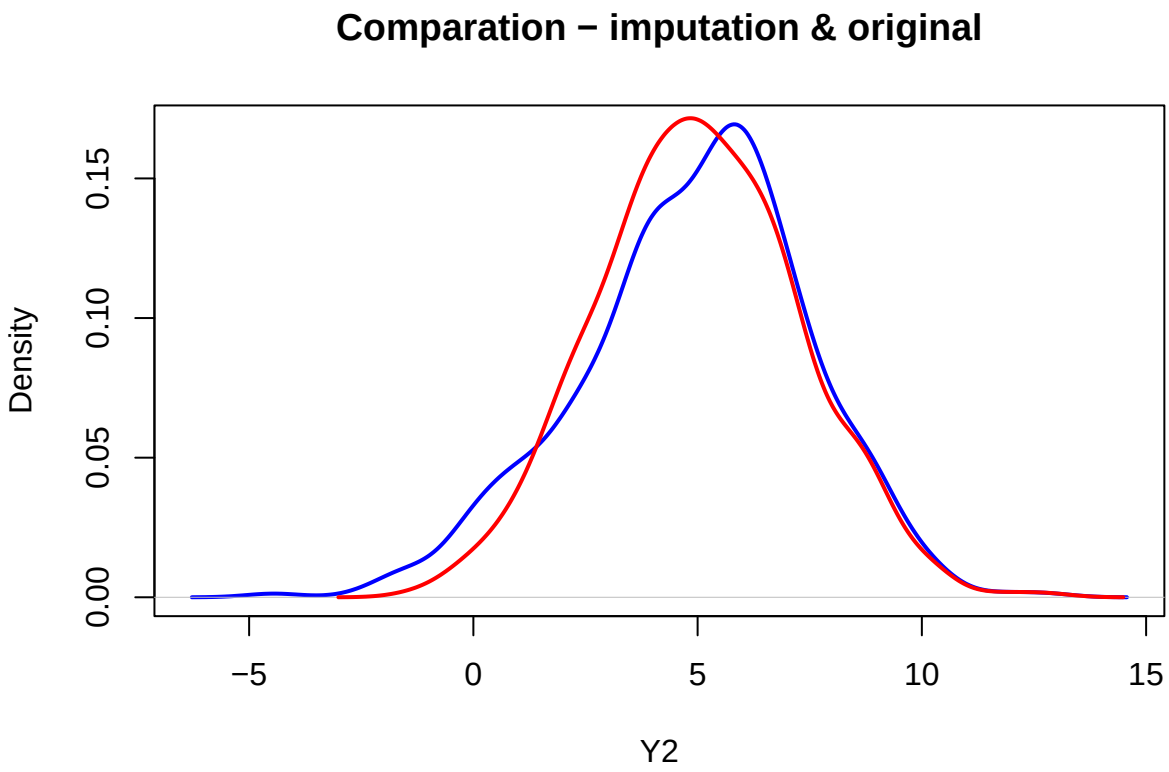
```
# get regression coefficient
beta <- coef(lm(y2~y1))
beta
```

```
## (Intercept)          y1
##    2.870139    1.999996
```

```
# generate correspond bias selement
set.seed(2)
z_bias <- rnorm(500,mean = 0, sd = sd(y2,na.rm = TRUE))

# impute missing y2 by stochastic regression imputation
y2_pre <- y2
for (i in y2_miss_index){
  y2_pre[i] <- beta[1] + y1[i]*beta[2] + z_bias[i]}
```

```
# draw marginal distribution plot
plot(density(y2_pre), lwd = 2, col = "blue", xlab = "Y2",
     main = "Comparation - imputation & original")
lines(density(y2_original), lwd = 2, col = "red")
```



There are 2 different curves in the upper graph represent the original (without missing) Y_2 (represents by the red curve) and Y_2 after we imputing those missing values (represents by the blue curve). We can see from 2 curves that 2 curves have similar density distribution, which means we get good imputation values by stochastic regression imputation this time and *seed(2)* bias elements.

Sub-Question (c)

1. Simulate the corresponding observed dataset ($a = 0, b = 2$)

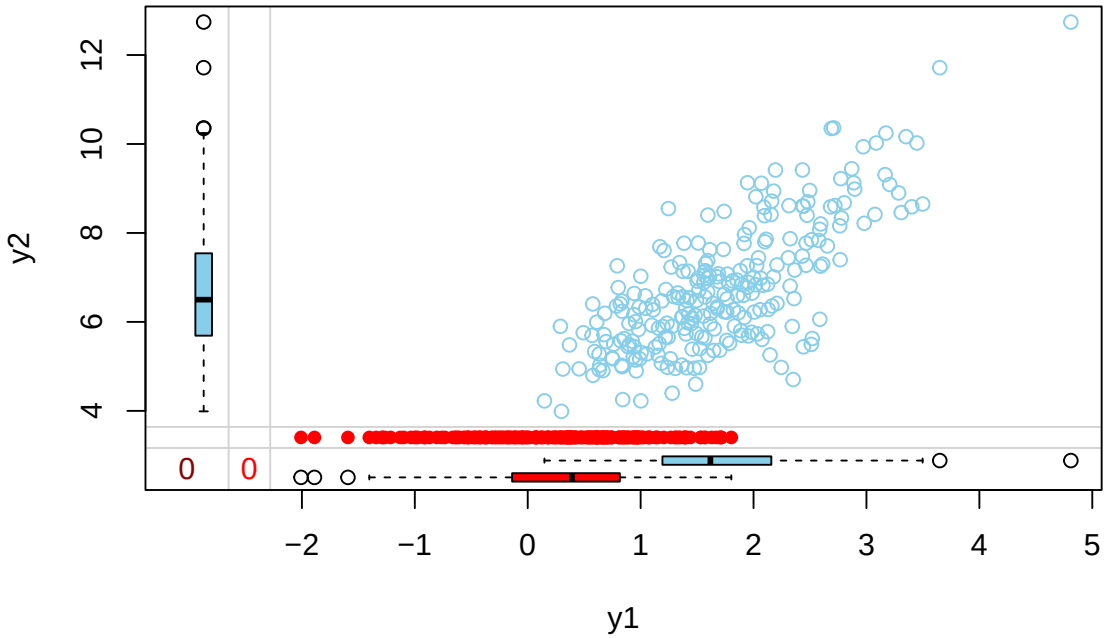
```
# simulate corresponding observed data (a = 2, b = 0)
a <- 0
b <- 2
# eveluate if y2 is missing
missing_evaluate <- (a*(y1-1) + b*(y2_original-5) + z3)
y2_miss_index <- which(missing_evaluate < 0)

# get 'real' y2
y2 <- y2_original
for (i in y2_miss_index){
  y2[i] <- NA}
```

2. Display the marginal distribution of Y_2 and evaluate the mechanism.

   - Comparing Y_1 and Y_2

```r
# display the margnial distribution(y1,y2)
require(VIM)
Y <- data.frame(y1,y2)
marginplot(Y)
```



The blue point in the largest area correspond to observations which both Y_1 and Y_2 are observed. The red dots below correspond to records which Y_1 is observed while Y_2 is missing. The red boxplot below represents the marginal distribution of Y_1 observation which the corresponding Y_2 are missing. The blue boxplot below represents the marginal distribution of Y_1 which the corresponding Y_2 are observed. The blue boxplot on the left represents the marginal distribution of Y_2.

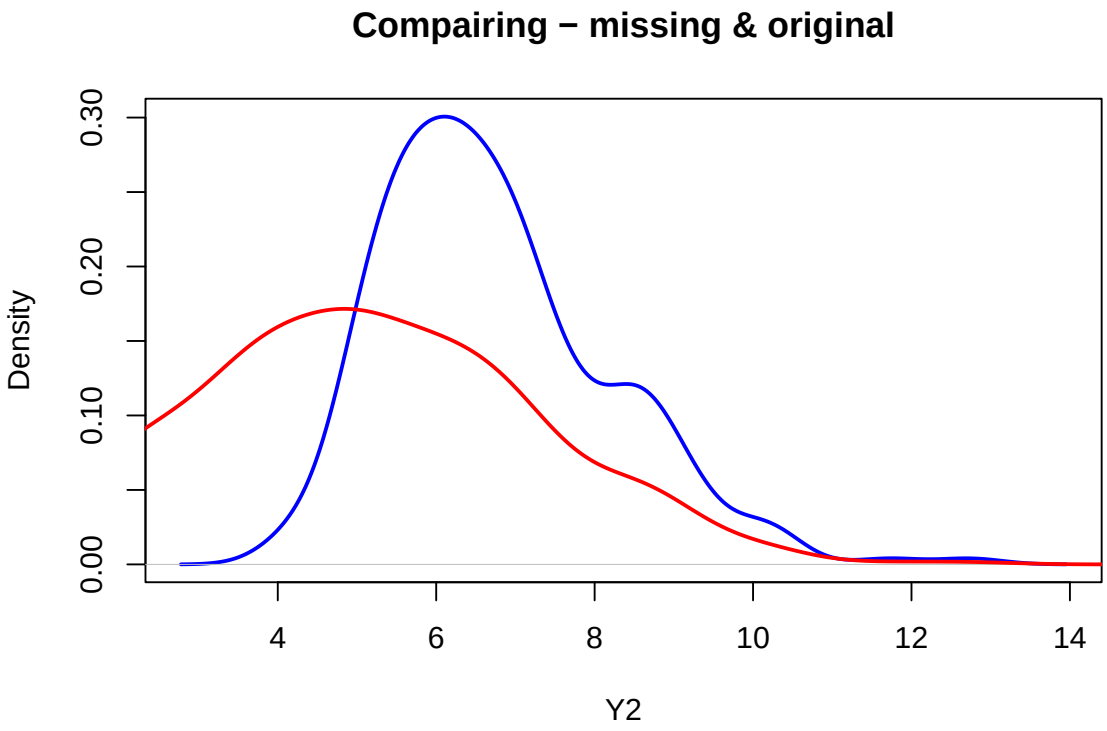
We can see from 2 boxplots below that the marginal distributions of Y_1 have difference when Y_2 is observed or missing. The mechanism is not MCAR, as if it is, the 2 distributions are expected to be identity.

- Comparing original Y_2 and 'real' Y_2 (after imposing)

```r
# y2_obs is a vector only contain y2 observed values
y2_obs <- c()
for (i in y2){
  if (!(is.na(i))){
    y2_obs <- c(y2_obs,i)
  }
}

# plot marginal distributions
plot(density(y2_obs), lwd = 2, col = "blue", xlab = "Y2",
```

```
     main = "Compairing - missing & original")
lines(density(y2_original), lwd = 2, col = "red")
```



We can see that the observed data of Y_2 (representing by the red curve) and the 'complete' data of Y_2 (representing by the blue curve) have different distributions.

In real situation, as we cannot observed the missing value, we cannot absolutely say that the mechanism is MNAR or MAR. In this graph, as 2 Y_2 have an obvious difference. Therefore, I hold the opinion that it has a higher probability to be MNAR than MAR.

As those 'missing values' are generated by ourselves by using $a \times (Y_1 - 1) + b \times (Y_2 - 5) + z_3$ where $a = 0, b = 2$, we will know that the missing of Y_2 have no relation with Y_1 and have relation with Y_2 itself. So, by this information, we can say that the mechanism is **MNAR**.

Sub-Question(d)

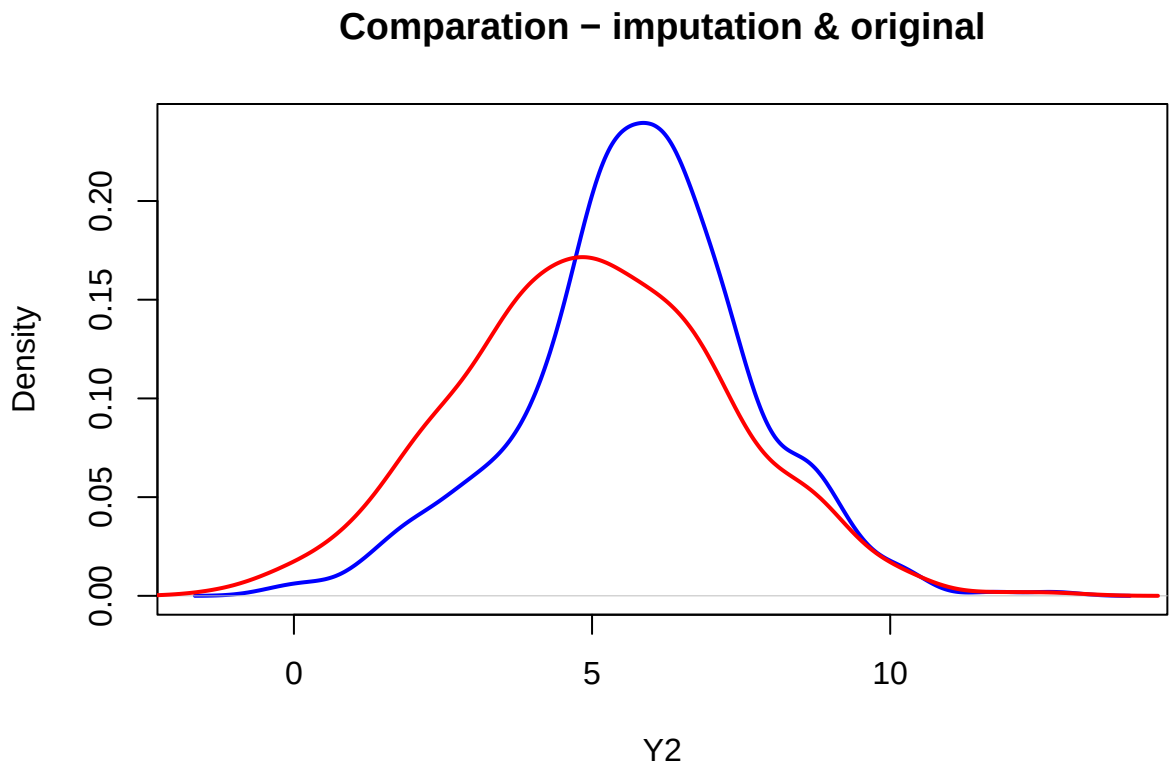
```
# get regression coefficient
beta <- coef(lm(y2~y1))

# generate correspond bias selement
set.seed(3)
z_bias <- rnorm(500,mean = 0, sd = sd(y2,na.rm = TRUE))

# impute missing y2 by stochastic regression imputation
y2_pre <- y2
```

```r
for (i in y2_miss_index){
  y2_pre[i] <- beta[1] + y1[i]*beta[2] + z_bias[i]}
```

```r
# draw marginal distribution plot
plot(density(y2_pre), lwd = 2, col = "blue", xlab = "Y2",
     main = "Comparation - imputation & original")
lines(density(y2_original), lwd = 2, col = "red")
```



There are 2 different curves in the upper graph represent the original (without missing) Y_2 (represents by the red curve) and Y_2 after we imputing those missing values (represents by the blue curve). We can see from 2 curves that the blue one is more concentrated on its mean value, which means the values,which are imposed by stochastic regression imputation and *seed(3)* random($\sim N(0, \sigma^2)$) bias elements, has a higher variance value than original data. And the mean value of Y_2 is also a little bit higher than original Y_2.

Comparing with the result in previous part (b), stochastic regression imputation may have better performance in MAR than MNAR.

Question 4

```r
# load data from dataset and have a brief view
load('databp.Rdata')
summary(databp)
```

```
##     logdose          bloodp        recovtime
##  Min.   :1.180   Min.   :52.00   Min.   : 7.00
##  1st Qu.:1.740   1st Qu.:61.00   1st Qu.:10.50
##  Median :1.880   Median :68.00   Median :16.00
##  Mean   :1.999   Mean   :67.24   Mean   :19.27
##  3rd Qu.:2.270   3rd Qu.:71.00   3rd Qu.:22.00
##  Max.   :2.780   Max.   :88.00   Max.   :60.00
##                                  NA's   :3
```

Sub-Question (a)

```
# PART (a): complete case analysis
# delete NA recording (carry out a complete dataset)
del_na <- databp
del_na <- na.omit(del_na)

# recover time
recov_mean_a <- mean(del_na$recovtime, na.rm = TRUE)
recov_mean_a
```

```
## [1] 19.27273
```

```
# recover time standard error
std.error <- function(x) sd(x)/sqrt(length(x))
recov_std_error_a <- std.error(del_na$recovtime)
recov_std_error_a
```

```
## [1] 2.603013
```

```
# correlations (recovery time & dose)
cor_rec_does_a <- cor(del_na$recovtime,
                      del_na$logdose, method = c("pearson"))
cor_rec_does_a
```

```
## [1] 0.2391256
```

```
# correlations (recovery time & blood pressure)
cor_rec_bp_a <- cor(del_na$recovtime,
                    del_na$bloodp, method = c("pearson"))
cor_rec_bp_a
```

```
## [1] -0.01952862
```

The mean value of the recovery time is **19.27**, the standard value of the recovery time is **2.603**. The pearson correlation between recover time and does is **0.239**, the pearson correlation between recover time and blood pressure is **-0.0195**.

Sub-Question(b)

```r
# PART (b): mean imputation
# using mean value of recovtime(observed) replace missing value
rep_mean_na <- databp
rep_mean_na$recovtime[is.na(rep_mean_na$recovtime)] <-
  mean(rep_mean_na$recovtime[!is.na(rep_mean_na$recovtime)])

# recover time mean
recov_mean_b <- mean(rep_mean_na$recovtime)
recov_mean_b
```

```
## [1] 19.27273
```

```r
# recover time standard error
recov_std_error_b <- std.error(rep_mean_na$recovtime)
recov_std_error_b
```

```
## [1] 2.284135
```

```r
# correlations (recovery time & dose)
cor_rec_does_b <- cor(rep_mean_na$recovtime,
                      rep_mean_na$logdose, method = c("pearson"))
cor_rec_does_b
```

```
## [1] 0.2150612
```

```r
# correlations (recovery time & blood pressure)
cor_rec_bp_b <- cor(rep_mean_na$recovtime,
                    rep_mean_na$bloodp, method = c("pearson"))
cor_rec_bp_b
```

```
## [1] -0.01934126
```

The mean value of the recovery time is **19.27**, the standard value of the recovery time is **2.284**. The pearson correlation between recover time and does is **0.215**, the pearson correlation between recover time and blood pressure is **-0.0193**.

Sub-Question(c)

```r
# PART (c): mean regression imputation method
# initialize dataset and locate those missing values' indexes
rep_reg_na <- databp
recover_miss_index <- which(is.na(databp$recovtime))

# calculate coefficients for regression(linear)
beta_4 <- coef(lm(databp$recovtime~databp$logdose + databp$bloodp))
beta_4
```

```
##    (Intercept) databp$logdose  databp$bloodp
##     15.2159065     11.4290287     -0.2769265
```

```r
# using calculated coefficients to get predict missing value and imputing them
for (i in recover_miss_index){
  rep_reg_na$recovtime[i] <- beta_4[1] +
    rep_reg_na$logdose[i]*beta_4[2] + rep_reg_na$bloodp[i]*beta_4[3]}

# calculate what the problem asked for
# recover time mean
recov_mean_c <- mean(rep_reg_na$recovtime, na.rm = TRUE)
recov_mean_c
```

```
## [1] 19.44428
```

```r
# recover time standard error
recov_std_error_c <- std.error(rep_reg_na$recovtime)
recov_std_error_c
```

```
## [1] 2.312845
```

```r
# correlations (recovery time & dose)
cor_rec_does_c <- cor(rep_reg_na$recovtime,
                      rep_reg_na$logdose, method = c("pearson"))
cor_rec_does_c
```

```
## [1] 0.2801835
```

```r
# correlations (recovery time & blood pressure)
cor_rec_bp_c <- cor(rep_reg_na$recovtime,
                    rep_reg_na$bloodp, method = c("pearson"))
cor_rec_bp_c
```

```
## [1] -0.0111364
```

The mean value of the recovery time is **19.44**, the standard value of the recovery time is **2.313**. The pearson correlation between recover time and does is **0.280**, the pearson correlation between recover time and blood pressure is **-0.0111**.

Sub-Question(d)

```r
# PART (d): stochastic regression imputation method
# initialize dataset and enough residual values
rep_sto_na <- databp
set.seed(4)
z_bias_4 <- rnorm(nrow(databp),0,sd = sd(databp$recovtime,na.rm = TRUE))

# using calculated coefficients to get predict missing value and imputing them
for (i in recover_miss_index){
  rep_sto_na$recovtime[i] <- beta_4[1] + rep_sto_na$logdose[i]*beta_4[2]
  + rep_sto_na$bloodp[i]*beta_4[3] + z_bias_4[i]}
```

```r
# calculate what the problem asked for
# recover time mean
recov_mean_d <- mean(rep_sto_na$recovtime, na.rm = TRUE)
recov_mean_d
```

```
## [1] 21.74831
```

```r
# recover time standard error
recov_std_error_d <- std.error(rep_sto_na$recovtime)
recov_std_error_d
```

```
## [1] 2.687974
```

```r
# correlations (recovery time & dose)
cor_rec_does_d <- cor(rep_sto_na$recovtime,
                      rep_sto_na$logdose, method = c("pearson"))
cor_rec_does_d
```

```
## [1] 0.3171839
```

```r
# correlations (recovery time & blood pressure)
cor_rec_bp_d <- cor(rep_sto_na$recovtime,
                    rep_sto_na$bloodp, method = c("pearson"))
cor_rec_bp_d
```

```
## [1] 0.03484167
```

The mean value of the recovery time is **21.75**, the standard value of the recovery time is **2.688**. The pearson correlation between recover time and does is **0.317**, the pearson correlation between recover time and blood pressure is **0.035**.

Sub-Question(e)

```r
# PART (e): predictive mean matching
# initialize dataset and a vector for predicted recover time value
PMM_NA <- databp
PMM_recover <- c()

# Use regression coefficients beta_4 in sub-question(c) to calculate predictied values
for(i in 1:length(databp$recovtime)){
  PMM_recover[i] <- beta_4[1] + rep_reg_na$logdose[i]*beta_4[2]
  + rep_reg_na$bloodp[i]*beta_4[3]
}

# locating observed recover time values' indexes
recover_obs_index <- which(!is.na(databp$recovtime))
# the missing values index is located befor: recover_miss_index

# compare missing element's predictive values with observed elements' predictived values
```

```
for(i in recover_miss_index){
  auxuaily <- Inf
  for(j in recover_obs_index){
    abs <- abs(PMM_recover[i]-PMM_recover[j])
    if(abs < auxuaily){
      auxuaily <- abs
      donor <- PMM_NA$recovtime[j]
      }
  }
  PMM_NA$recovtime[i] <- donor
}
# show imputation result
PMM_NA
```

```
##    logdose bloodp recovtime
## 1     2.26     66         7
## 2     1.81     52        10
## 3     1.78     72        18
## 4     1.54     67        14
## 5     2.06     69        10
## 6     1.74     71        13
## 7     2.56     88        21
## 8     2.29     68        12
## 9     1.80     59         9
## 10    2.32     73        12
## 11    2.04     68        20
## 12    1.88     58        31
## 13    1.18     61        23
## 14    2.08     68        22
## 15    1.70     69        13
## 16    1.74     55         9
## 17    2.70     73        39
## 18    1.90     56        28
## 19    2.78     83        12
## 20    2.27     67        60
## 21    1.74     84        10
## 22    2.62     68        21
## 23    1.80     64        22
## 24    1.81     60        21
## 25    1.58     62        14
```

```
# calculate what the problem asked for
# recover time mean
recov_mean_e <- mean(PMM_NA$recovtime, na.rm = TRUE)
recov_mean_e
```

```
## [1] 18.84
```

```
# recover time standard error
recov_std_error_e <- std.error(PMM_NA$recovtime)
recov_std_error_e
```

```
## [1] 2.312776
```

```r
# correlations (recovery time & dose)
cor_rec_does_e <- cor(PMM_NA$recovtime,
                      PMM_NA$logdose, method = c("pearson"))
cor_rec_does_e
```

```
## [1] 0.223028
```

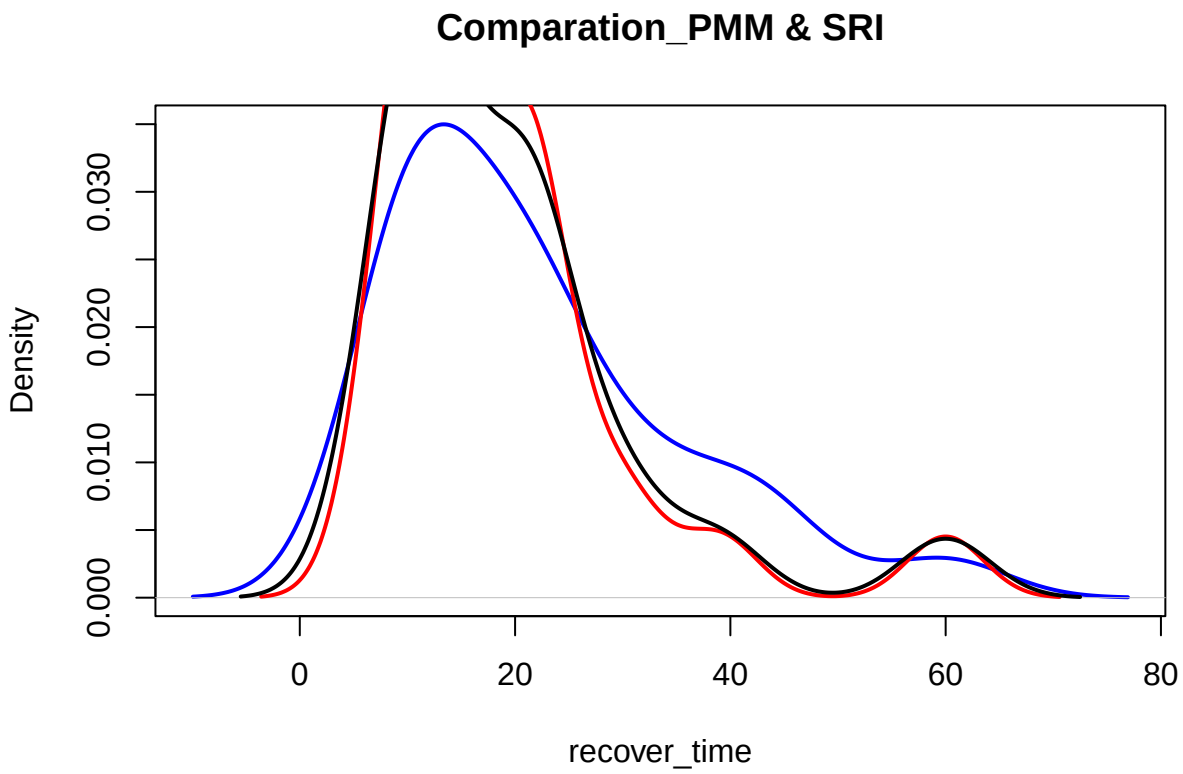
```r
# correlations (recovery time & blood pressure)
cor_rec_bp_e <- cor(PMM_NA$recovtime,
                    PMM_NA$bloodp, method = c("pearson"))
cor_rec_bp_e
```

```
## [1] -0.03517248
```

The mean value of the recovery time is **18.84**, the standard value of the recovery time is **2.313**. The pearson correlation between recover time and does is **0.223**, the pearson correlation between recover time and blood pressure is **-0.035**.

Sub-Question(f)

```r
# draw marginal distribution plot
plot(density(rep_sto_na$recovtime), lwd = 2, col = "blue",
     xlab = "recover_time", main = "Comparation_PMM & SRI")
lines(density(PMM_NA$recovtime), lwd = 2, col = "red")
lines(density(del_na$recovtime),lwd = 2, col = "black")
```



The upper graph shows 2 curves which the red one represents distribution of recover time after using MMP and the blue one represents distribution after using SRI. We can observe that by using PMM, the distribution is more smoothness while the curve fluctuating more after using SRI. I add a black curve to represent the distribution of recover time in complete data analysis, which means data in this curve are all observed real value. We can see that the red curve(PMM) are more similar to the black curve(real) comparing with the blue one(SRI).

Therefore I conclude that the advantage of PMM over SRI is that it can produce more realistic and accurate imputations by directly matching the imputed values to the observed values in the dataset. Specifically, PMM works by identifying the observed values in the dataset that are closest to the predicted value from the regression model for each missing value, based on the values of the other subjects' values in the model. This process ensures that the imputed values are drawn from the same distribution as the observed values in the dataset and better captures the variability and distribution of the data.However, SRI estimates the missing values based on the regression model and then adds random bias to the predicted values, which may not always produce imputed values that are conform to realistic comparing with PMM. Based on this, another advantage of PMM is that it may handle complex or nonlinear relationships more accurate as SRI assumes that the relationship is linear.