Assignment 1

Zekai Li S2040608

Question 1

(a) **My solution**

The answer is (ii).

For ALQ is MCAR, the distribution of ALQ being missing is identical to that being not missing. And there would be the same probability of ALQ being missing for both those with ALQ=Yes and ALQ=No. And that is 0.3 from the description of the problem.

(b) **My solution** The answer is (ii).

For ALQ is MAR and that being missing depends only on the observed values, in this case, gender means. So after adjusting for gender, the probability of ALQ being missing is independent of Yes/No value of ALQ.

(c) **My solution** The answer is (iii).

For ALQ is MAR given gender, so the ALQ being missing is not only dependent on the gender but also dependent on the Yes/No value of ALQ. So for two different gender groups, the probability of ALQ being missing is uncertain.

Question 2

My solution

- Largest case: there would be 90 subjects when the missing values happens for each variable at the same time. Then there are only 10% subjects would be discarded.
- Smallest case: there would be 0 subjects when there would be at most one variable being missing for each subject. So there woulbe be $10\times(100\times 10\%) = 100$ subjects being discarded. Thus the smallest subsample is 0.

Question 3

(a) **My solution**

This mechanism is MAR. Since the missingness indicator:

$$\begin{aligned} Pr(\mathbf{R} < 0|Y_1, Y_2) &= Pr(a \times (Y_1 - 1) + b \times (Y_2 - 5) + Z_3 < 0|Y_1, Y_2) \\ &= Pr(Z_3 < -a \times (Y_1 - 1) - b \times (Y_2 - 5)) \end{aligned}$$

is not relevant to Y_2 when $b = 0$, the mechanism is not MCAR. and that probability is relevant to Y_1 since $a \neq 0$, so the mechanism is not MNAR.

And the marginal distributions of Y_2 for two datasets have been depicted blow. From the chart, the mean of the two datasets is obviously different and std of obeserved data would also be smaller than complete ones after comparing the density values of the two datasets. Thus, we can conclude that the missing mechanism is not MCAR. And from the chart the distribution for observed dataset is still similar to norm distribution, which means the distributions of Y_2 being missing and being observed are similar, indicating the missing mechanism would probably not be MNAR.

```
set.seed(42)
complete_data =
  data.frame('Z1' = rnorm(500), 'Z2' = rnorm(500)) %>%
  tibble() %>%
  summarise(
    Y1 = 1+Z1,
    Y2 = 5+2*Z1+Z2
  )

observed_data =
  complete_data %>%
  mutate(
    Z3 = rnorm(500),
    Missing = 2*(Y1-1) + 0*(Y2-5) + Z3,
    Y2 = map2_dbl(Y2,Missing,~if_else(.y<0, as.double(NA), .x))
  ) %>% select(Y1,Y2) %>% rename(Y2_missing = 'Y2')

complete_data
```

```
## # A tibble: 500 x 2
##       Y1    Y2
##    <dbl> <dbl>
##  1 2.37   8.77
##  2 0.435  4.79
##  3 1.36   5.72
##  4 1.63   6.40
##  5 1.40   5.09
##  6 0.894  4.59
##  7 2.51   6.99
##  8 0.905  3.84
##  9 3.02   7.82
## 10 0.937  5.71
## # ... with 490 more rows
```

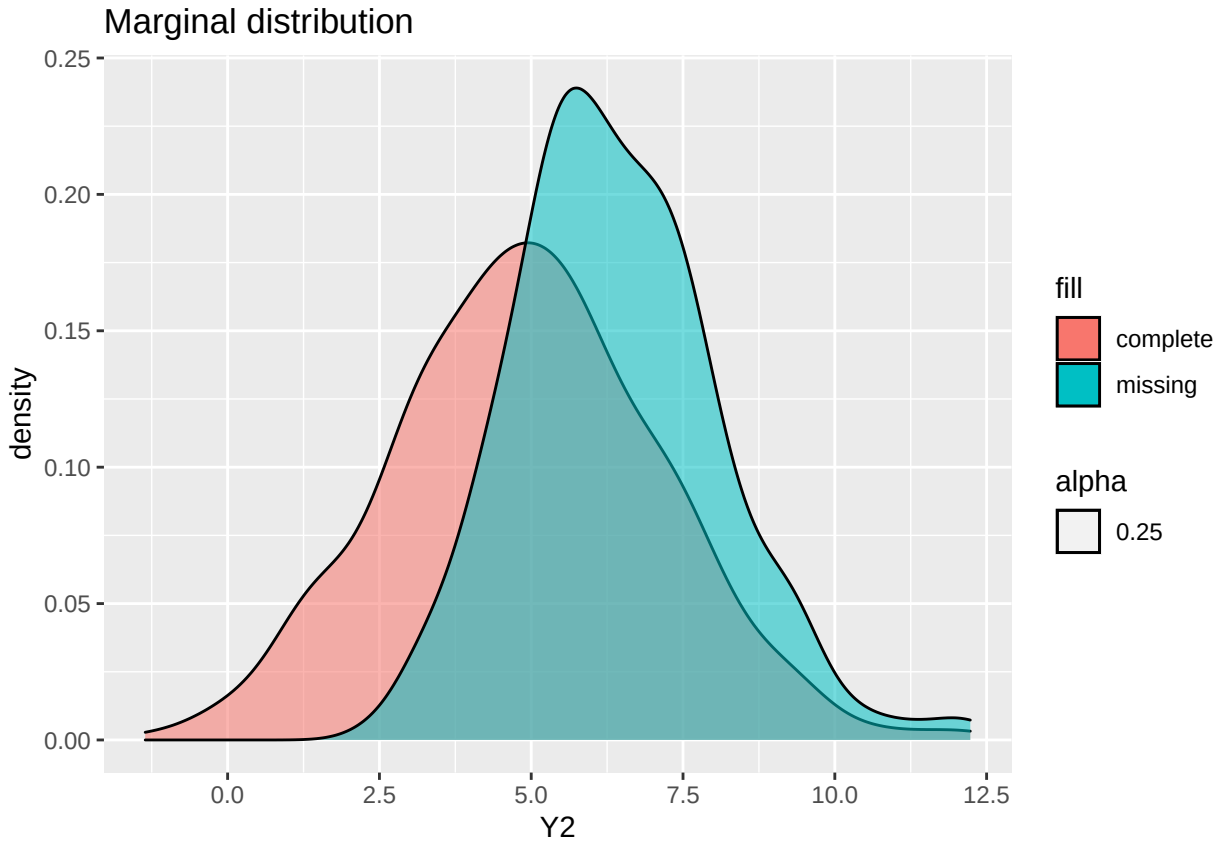
```
observed_data
```

```
## # A tibble: 500 x 2
##       Y1 Y2_missing
##    <dbl>      <dbl>
##  1 2.37        8.77
##  2 0.435      NA
##  3 1.36        5.72
##  4 1.63        6.40
```

```
##  5 1.40       NA
##  6 0.894      NA
##  7 2.51        6.99
##  8 0.905      NA
##  9 3.02        7.82
## 10 0.937       5.71
## # ... with 490 more rows
```

```r
complete_data %>%
  full_join(., observed_data, by = 'Y1') %>%
  ggplot() + geom_density(
    mapping = aes(
      x = Y2,
      fill = 'complete', alpha = 0.25
    )
  ) + geom_density(
    mapping = aes(
      x = Y2_missing,
      fill = 'missing', alpha = 0.25
    )
  ) +
  labs(
    title = 'Marginal distribution',
    x = 'Y2'
  )
```

```
## Warning: Removed 256 rows containing non-finite values (stat_density).
```



(*b*) **My solution**

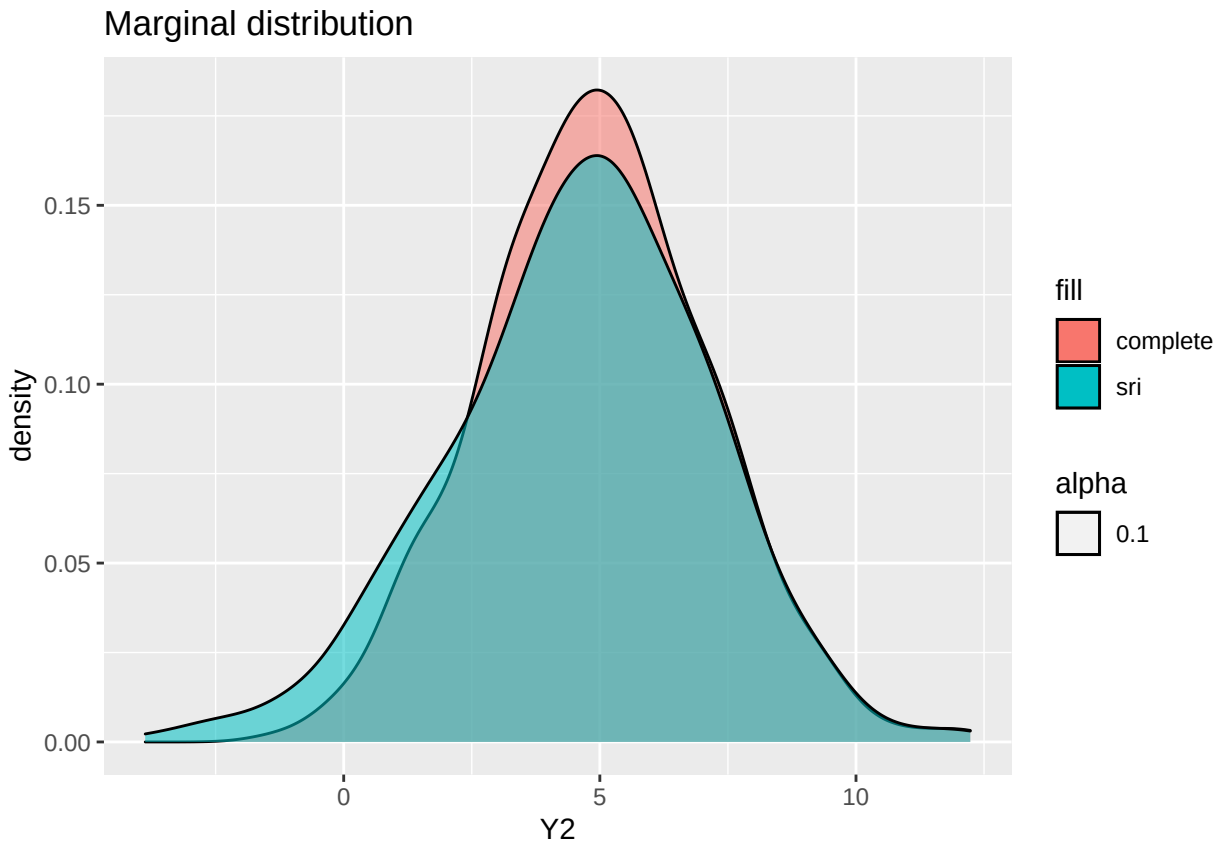
After single imputation, the missing values of Y_2 has been made up by the values learnt from the data Y_1. Since Y_1 being missing is only dependent on Y_1, the marginal density of Y_2 is similar to complete data. That means the SRI single imputation methods play well in this case.

```
set.seed(42)
sri_data =
  observed_data %>%
  mutate(
    predicted = lm(Y2_missing ~ Y1) %>%
      predict(newdata = observed_data),
    stochastic_values = lm(Y2_missing ~ Y1) %>%
      sigma() %>% rnorm(500,0, .),
    predict = predicted + stochastic_values,
    Y2_sri = map2_dbl(Y2_missing, predict, ~ if_else(is.na(.x), .y, .x))
  )%>% select(Y1, Y2_sri)

sri_data
```

```
## # A tibble: 500 x 2
##       Y1 Y2_sri
##    <dbl>  <dbl>
##  1 2.37    8.77
##  2 0.435   3.37
##  3 1.36    5.72
##  4 1.63    6.40
##  5 1.40    6.26
##  6 0.894   4.73
##  7 2.51    6.99
##  8 0.905   4.77
##  9 3.02    7.82
## 10 0.937   5.71
## # ... with 490 more rows
```

```
complete_data %>%
  full_join(., sri_data, by = 'Y1') %>%
  ggplot() + geom_density(
    mapping = aes(
      x = Y2,
      fill = 'complete', alpha = 0.1
    )
  ) + geom_density(
    mapping = aes(
      x = Y2_sri,
      fill = 'sri', alpha = 0.1
    )
  ) +
  labs(
    title = 'Marginal distribution',
    x = 'Y2'
  )
```



(*c*) **My solution**

This mechanism is MNAR. Since the missingness indicator:

$$\begin{aligned} Pr(\mathbf{R} < 0|Y_1, Y_2) &= Pr(a \times (Y_1 - 1) + b \times (Y_2 - 5) + Z_3 < 0|Y_1, Y_2) \\ &= Pr(Z_3 < -a \times (Y_1 - 1) - b \times (Y_2 - 5)) \end{aligned}$$

is not relevant to Y_1 when $a = 0$, the mechanism is not MAR, but that probability is relevant to Y_2 since $b \neq 0$, so the mechanism is MNAR.

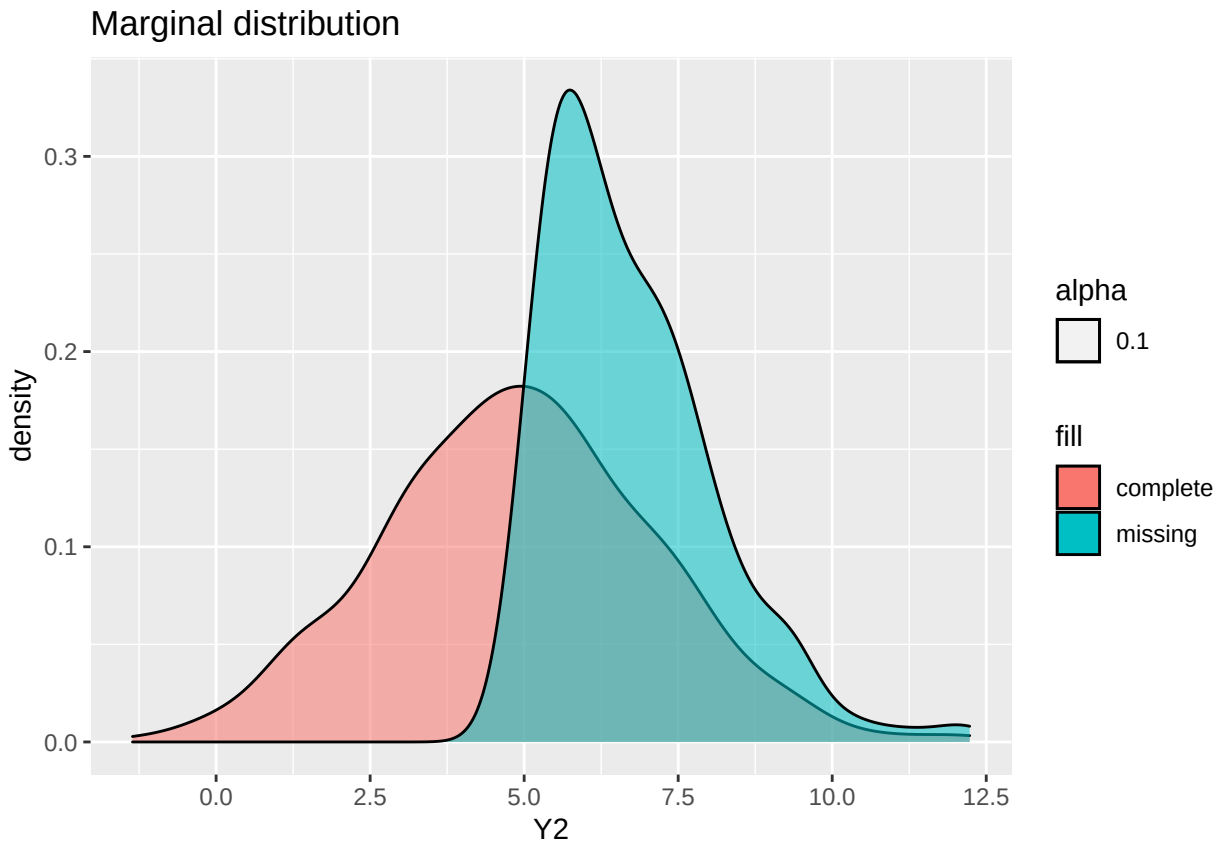
And from the chart displayed below, the distribution of Y_2 being missing has a larger mean an smaller std which means the missing mechanism would not like to be MCAR. And the distribution of Y_2 being missing is likely to be asysmmetric. So the distribution is different between the missing values and complete values, indicating the missing mechansim is MNAR.

```
set.seed(42)
observed_data =
  complete_data %>%
  mutate(
    Z3 = rnorm(500),
    Missing = 0*(Y1-1) + 2*(Y2-5) + Z3,
    Y2_missing = map2_dbl(Y2,Missing,~if_else(.y<0, as.double(NA), .x))
  ) %>%
  select(Y1,Y2_missing)
observed_data
```

```
## # A tibble: 500 x 2
##       Y1 Y2_missing
##    <dbl>      <dbl>
##  1 2.37        8.77
##  2 0.435      NA
##  3 1.36        5.72
##  4 1.63        6.40
##  5 1.40        5.09
##  6 0.894      NA
##  7 2.51        6.99
##  8 0.905      NA
##  9 3.02        7.82
## 10 0.937       5.71
## # ... with 490 more rows
```

```
complete_data %>%
  full_join(., observed_data, by = 'Y1') %>%
  ggplot() + geom_density(
    mapping = aes(
      x = Y2,
      fill = 'complete', alpha = 0.1
    )
  ) + geom_density(
    mapping = aes(
      x = Y2_missing,
      fill = 'missing', alpha = 0.1
    )
  ) +
  labs(
    title = 'Marginal distribution',
    x = 'Y2'
  )
```

```
## Warning: Removed 262 rows containing non-finite values (stat_density).
```



(*d*) **My solution** After Stochastic Regression Imputation, the missing values of Y_2 has been made up by the values learnt from the data Y_1. But the missing mechanism shows that the missing values of Y_2 is not influenced by the values of Y_1, so the result of simulation is not as well as the previous question does. But the values of Y_2 still has some relationship with Y_1, which also makes the simulation being more similar to the complete data.

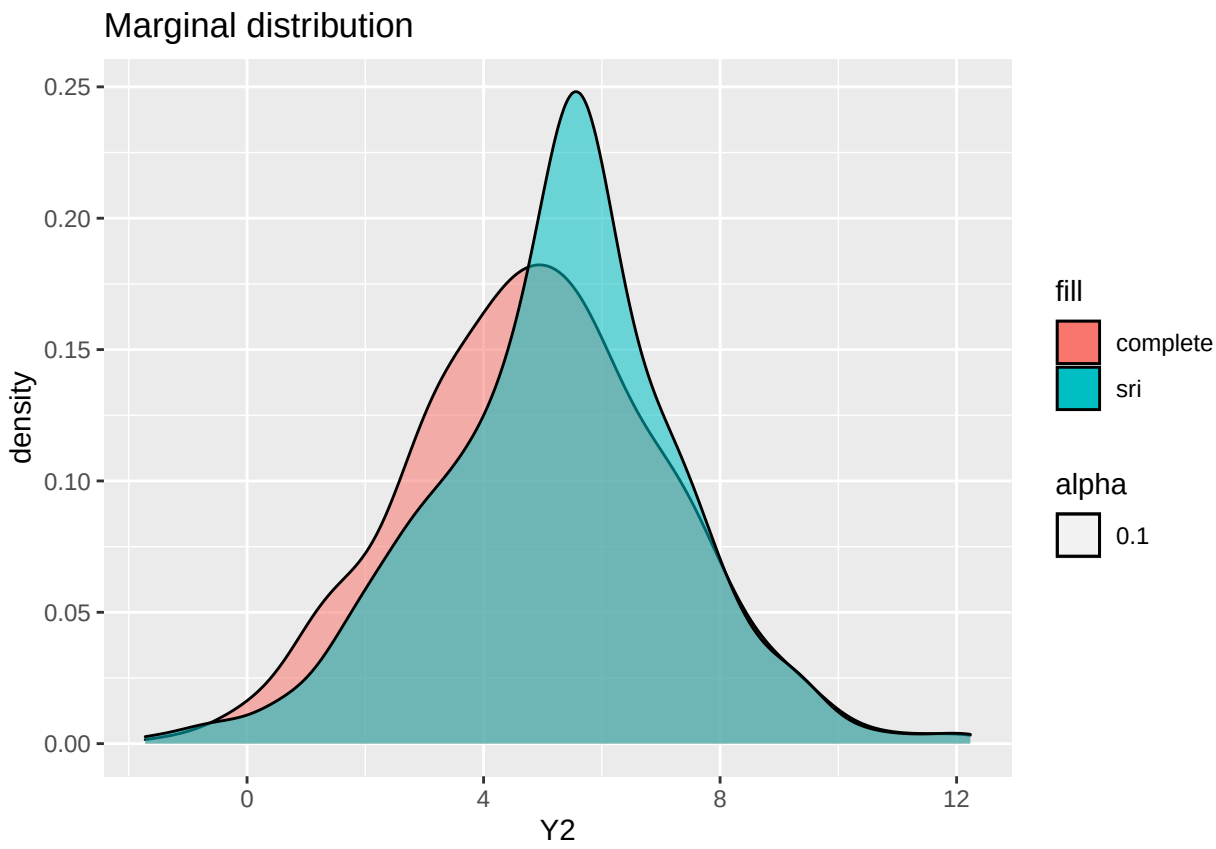
```
set.seed(42)
sri_data =
  observed_data %>%
  mutate(
    predicted = lm(Y2_missing ~ Y1) %>%
      predict(newdata = observed_data),
    stochastic_values = lm(Y2_missing ~ Y1) %>%
      sigma() %>% rnorm(500,0, .),
    predict = predicted + stochastic_values,
    Y2_sri = map2_dbl(Y2_missing, predict, ~ if_else(is.na(.x), .y, .x))
  )%>% select(Y1, Y2_sri)

sri_data
```

```
## # A tibble: 500 x 2
##       Y1 Y2_sri
##    <dbl>  <dbl>
##  1 2.37    8.77
##  2 0.435   4.24
##  3 1.36    5.72
```

```
##  4 1.63    6.40
##  5 1.40    5.09
##  6 0.894   5.37
##  7 2.51    6.99
##  8 0.905   5.40
##  9 3.02    7.82
## 10 0.937   5.71
## # ... with 490 more rows
```

```
complete_data %>%
  full_join(., sri_data, by = 'Y1') %>%
  ggplot() + geom_density(
    mapping = aes(
      x = Y2,
      fill = 'complete', alpha = 0.1
    )
  ) + geom_density(
    mapping = aes(
      x = Y2_sri,
      fill = 'sri', alpha = 0.1
    )
  ) +
  labs(
    title = 'Marginal distribution',
    x = 'Y2'
  )
```



Question 4

```
load('databp.Rdata')
databp = databp %>%
  tibble()

databp
```

```
## # A tibble: 25 x 3
##    logdose bloodp recovtime
##      <dbl>  <dbl>     <dbl>
##  1    2.26     66         7
##  2    1.81     52        10
##  3    1.78     72        18
##  4    1.54     67        NA
##  5    2.06     69        10
##  6    1.74     71        13
##  7    2.56     88        21
##  8    2.29     68        12
##  9    1.8      59         9
## 10    2.32     73        NA
## # ... with 15 more rows
```

(*a*) My solution

after CCA, the mean is 19.27, the standard deviation is 12.21, the pearson corralation between simulated recover time and logdose is 0.239,the pearson corralation between simulated recovtime and blood pressure is -0.0195.

```
databp_CCA =
  databp%>%
  filter(
    !is.na(databp$recovtime)
  )

setNames(
  c(mean(databp_CCA$recovtime, na.rm = TRUE),
  sd(databp_CCA$recovtime),
  cor(databp_CCA$recovtime, databp_CCA$logdose, method = "pearson"),
  cor(databp_CCA$recovtime, databp_CCA$bloodp, method = "pearson")), c('mean','std','pearson cor dose',
)
```

```
##             mean              std pearson cor dose person cor blood
##      19.27272727      12.20921517       0.23912558      -0.01952862
```

(*b*) My solution

after MI, the mean is 19.27, the standard deviation is 11.42, the pearson corralation between simulated recover time and logdose is 0.215,the pearson corralation between simulated recovtime and blood pressure is -0.0193.

```
databp_MI =
  databp%>%
  mutate(
    MI = ifelse(is.na(databp$recovtime), mean(databp$recovtime, na.rm = TRUE), databp$recovtime)
  )

setNames(
  c(mean(databp_MI$recovtime, na.rm = TRUE),
  sd(databp_MI$MI),
  cor(databp_MI$MI, databp_MI$logdose, method = "pearson"),
  cor(databp_MI$MI, databp_MI$bloodp, method = "pearson")), c('mean','std','pearson cor dose','person co
)
```

```
##             mean              std pearson cor dose person cor blood
##      19.27272727      11.42067503       0.21506117      -0.01934126
```

(*c*) **My solution**

after RI, the mean is 19.44, the standard deviation is 11.56, the pearson corralation between simulated recover time and logdose is 0.280,the pearson corralation between simulated recovtime and blood pressure is -0.0111.

```
databp_RI =
  databp%>%
  mutate(
    predict = lm(recovtime ~ logdose+bloodp) %>%
      predict(newdata = databp),
    RI = map2_dbl(recovtime, predict, ~ if_else(is.na(.x), .y, .x))
    ) %>% select(logdose, bloodp, RI)

setNames(
  c(mean(databp_RI$RI, na.rm = TRUE),
  sd(databp_RI$RI),
  cor(databp_RI$RI, databp_RI$logdose, method = "pearson"),
  cor(databp_RI$RI, databp_RI$bloodp, method = "pearson")), c('mean','std','pearson cor dose','person co
)
```

```
##             mean              std pearson cor dose person cor blood
##       19.4442848       11.5642244        0.2801835       -0.0111364
```

(*d*) **My solution**

after SRI, the mean is 18.85, the standard deviation is 11.82, the pearson corralation between simulated recover time and logdose is 0.119,the pearson corralation between simulated recovtime and blood pressure is -0.0200.

```
set.seed(42)
databp_SRI =
  databp%>%
  mutate(
    predicted = lm(recovtime ~ logdose+bloodp) %>%
```

```
      predict(newdata = databp),
    stochastic_values = lm(recovtime ~ logdose+bloodp) %>%
      sigma() %>% rnorm(nrow(databp),0, .),
    predict = predicted + stochastic_values,
    SRI = map2_dbl(recovtime, predict, ~ if_else(is.na(.x), .y, .x))
    ) %>% select(logdose, bloodp, SRI)

setNames(
  c(mean(databp_SRI$SRI, na.rm = TRUE),
  sd(databp_SRI$SRI),
  cor(databp_SRI$SRI, databp_SRI$logdose, method = "pearson"),
  cor(databp_SRI$SRI, databp_SRI$bloodp, method = "pearson")), c('mean','std','pearson cor dose','perso
)
```

```
##             mean              std pearson cor dose person cor blood
##      18.85068753      11.82115644       0.11855870      -0.02004183
```

(*e*) **My solution**

after Predicted Meaning Match, the mean is 19.40, the standard deviation is 12.35, the pearson corralation between simulated recover time and logdose is 0.0518,the pearson corralation between simulated recovtime and blood pressure is -0.0384s.

```
set.seed(42)

databp_PMM =
  databp %>%
  mutate(
    fitted = lm(recovtime ~ logdose+bloodp) %>%
      predict(newdata = databp),
    stochastic_values = lm(recovtime ~ logdose+bloodp) %>%
      sigma() %>% rnorm(nrow(databp),0, .),
    predicted = fitted + stochastic_values
    ) %>% select(logdose, bloodp, recovtime, predicted)

## doing PMM
mean_nonrespondent = databp_PMM$predicted[which(is.na(databp_PMM$recovtime))]
mean_respondent = databp_PMM$predicted[which(!is.na(databp_PMM$recovtime))]

result = matrix(0,3,22); mean_index = c()
rownames(result) <- names(mean_nonrespondent)
colnames(result) <- names(mean_respondent)

for(i in names(mean_nonrespondent)){
  for (j in names(mean_respondent)){
    result[i,j] = (mean_nonrespondent[i] - mean_respondent[j])**2
  }
  mean_index[i] =  names(which(result[i,] == min(result[i,])))
}

## attribute the simulated values into the dataframe
databp_PMM$recovtime[as.integer(names(mean_index))] = databp_PMM$recovtime[as.integer(mean_index)]
```

```r
## calculate basis statistics
setNames(
  c(mean(databp_PMM$recovtime, na.rm = TRUE),
  sd(databp_PMM$recovtime),
  cor(databp_PMM$recovtime, databp_PMM$logdose, method = "pearson"),
  cor(databp_PMM$recovtime, databp_PMM$bloodp, method = "pearson")), c('mean','std','pearson cor dose',
)
```

```
##             mean              std pearson cor dose person cor blood
##      19.40000000      12.34571451       0.05180814      -0.03844357
```

(f) My solution

- Advantages: Through PMM, the simulated values come from the original data which are potentially consistent. However, Stochastic Regression Imputation replaced the NA values by the estimate values over Stochastic Regression which may bring larger errors to the dataframe.
- Problems: ThroughPearson Mean Matching imputation, the predictive mean is only used for matching, and that makes it less sensitive to model misspecification.