exercise6 self solution

Xiang Li

2024/3/18

Part A

1

```
# set.seed(519)
# x = seq(0.001, 1, by=0.001)
# e = rnorm(1000, mean = 0, sd = 1)
# y =  2 + 5*x + e
# dat = data.frame(X=x, e=e, Y=y)
# write.csv(dat, file = 'dat.csv', row.names = FALSE)
```

```
dat = read.csv('dat.csv')
```

2

```
lr = lm(Y ~ X, data = dat)
print(summary(lr))
```

```
##
## Call:
## lm(formula = Y ~ X, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.00791 -0.69745 -0.03184  0.67755  2.98928
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.95691    0.06383   30.66   <2e-16 ***
## X            5.11890    0.11047   46.34   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.008 on 998 degrees of freedom
## Multiple R-squared:  0.6827, Adjusted R-squared:  0.6824
## F-statistic:  2147 on 1 and 998 DF,  p-value: < 2.2e-16
```

The estimated parameters are accurate.

3

a

```
set.seed(519)
dat$miss = rbinom(1000, size = 1, prob = 0.5)
dat$Y_MCAR = ifelse(dat$miss == 1, NA, dat$Y)
```

b

```
lr_MCAR = lm(Y_MCAR ~ X, data = dat, )
print(summary(lr_MCAR))
```

```
##
## Call:
## lm(formula = Y_MCAR ~ X, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.96276 -0.66732 -0.01346  0.66040  2.84857
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.91275    0.09096   21.03   <2e-16 ***
## X            5.11342    0.15603   32.77   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.007 on 502 degrees of freedom
##   (496 observations deleted due to missingness)
## Multiple R-squared:  0.6815, Adjusted R-squared:  0.6808
## F-statistic:  1074 on 1 and 502 DF,  p-value: < 2.2e-16
```

The estimated parameters are accurate, but the standard errors become larger.

c

```
library(mice)
```

```
##
## Attaching package: ’mice’

## The following object is masked from ’package:stats’:
##
##     filter

## The following objects are masked from ’package:base’:
##
##     cbind, rbind
```

```r
imp = mice(dat[, c('X', 'Y_MCAR')], method="norm", m=5)
```

```
## 
##  iter imp variable
##   1   1  Y_MCAR
##   1   2  Y_MCAR
##   1   3  Y_MCAR
##   1   4  Y_MCAR
##   1   5  Y_MCAR
##   2   1  Y_MCAR
##   2   2  Y_MCAR
##   2   3  Y_MCAR
##   2   4  Y_MCAR
##   2   5  Y_MCAR
##   3   1  Y_MCAR
##   3   2  Y_MCAR
##   3   3  Y_MCAR
##   3   4  Y_MCAR
##   3   5  Y_MCAR
##   4   1  Y_MCAR
##   4   2  Y_MCAR
##   4   3  Y_MCAR
##   4   4  Y_MCAR
##   4   5  Y_MCAR
##   5   1  Y_MCAR
##   5   2  Y_MCAR
##   5   3  Y_MCAR
##   5   4  Y_MCAR
##   5   5  Y_MCAR
```

```r
lr_imp_MCAR = with(imp, lm(Y_MCAR ~ X))
print(summary(lr_imp_MCAR))
```

```
## # A tibble: 10 x 6
##    term        estimate std.error statistic   p.value  nobs
##    <chr>          <dbl>     <dbl>     <dbl>     <dbl> <int>
##  1 (Intercept)     1.92    0.0631      30.5 1.49e-144  1000
##  2 X               5.08    0.109       46.4 1.06e-251  1000
##  3 (Intercept)     1.85    0.0669      27.7 1.22e-125  1000
##  4 X               5.22    0.116       45.1 3.52e-243  1000
##  5 (Intercept)     1.87    0.0631      29.7 3.72e-139  1000
##  6 X               5.12    0.109       46.9 2.50e-254  1000
##  7 (Intercept)     1.87    0.0616      30.4 7.19e-144  1000
##  8 X               5.20    0.107       48.8 2.74e-266  1000
##  9 (Intercept)     1.94    0.0629      30.9 2.41e-147  1000
## 10 X               5.01    0.109       46.0 7.26e-249  1000
```

```r
print(summary(pool(lr_imp_MCAR)))
```

```
##          term estimate  std.error statistic       df      p.value
## 1 (Intercept) 1.891270 0.07600773   24.8826 41.52637 1.423290e-26
## 2           X 5.124404 0.14709025   34.8385 19.84682 2.840948e-19
```

```r
dat_imp = complete(imp, "long", inc = TRUE)
```

The estimated parameters are accurate and standard errors are similar with the ture values.

4

a

```r
dat$Y_MAR = ifelse(dat$X > 0.5, NA, dat$Y)
```

b

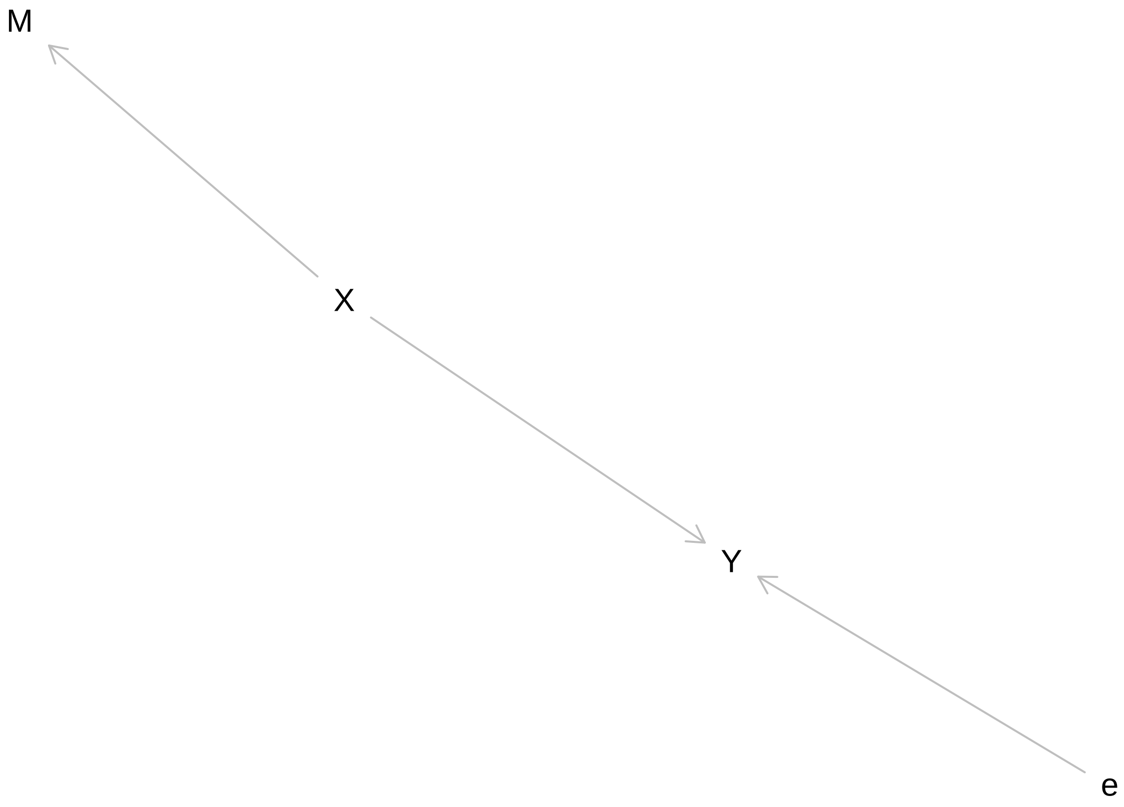
This is missing at random mechanism. Because the missing of Y is dependent on X.

c

```r
library(dagitty)
```

```r
set.seed(519)
g = dagitty('dag {
  X [exposure]
  Y [outcome]
  X -> { M Y }
  e  -> Y
  }')
plot(g)
```

```
## Plot coordinates for graph not supplied! Generating coordinates, see ?coordinates for how to set you
```



d

```r
lr_MAR = lm(Y_MAR ~ X, data = dat)
print(summary(lr_MAR))
```

```
##
## Call:
## lm(formula = Y_MAR ~ X, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.02601 -0.63225 -0.03613  0.63953  2.82916
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.00300    0.08794   22.78   <2e-16 ***
## X            4.96340    0.30419   16.32   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.9818 on 498 degrees of freedom
##   (500 observations deleted due to missingness)
## Multiple R-squared:  0.3484, Adjusted R-squared:  0.3471
## F-statistic: 266.2 on 1 and 498 DF,  p-value: < 2.2e-16
```

The coefficients are accurate and standard errors become larger.

e

```
imp = mice(dat[, c('X', 'Y_MAR')], method="norm", m=5)
```

```
##
##  iter imp variable
##   1   1  Y_MAR
##   1   2  Y_MAR
##   1   3  Y_MAR
##   1   4  Y_MAR
##   1   5  Y_MAR
##   2   1  Y_MAR
##   2   2  Y_MAR
##   2   3  Y_MAR
##   2   4  Y_MAR
##   2   5  Y_MAR
##   3   1  Y_MAR
##   3   2  Y_MAR
##   3   3  Y_MAR
##   3   4  Y_MAR
##   3   5  Y_MAR
##   4   1  Y_MAR
##   4   2  Y_MAR
##   4   3  Y_MAR
##   4   4  Y_MAR
##   4   5  Y_MAR
##   5   1  Y_MAR
##   5   2  Y_MAR
##   5   3  Y_MAR
##   5   4  Y_MAR
##   5   5  Y_MAR
```

```
lr_imp_MAR = with(imp, lm(Y_MAR ~ X))
print(summary(lr_imp_MAR))
```

```
## # A tibble: 10 x 6
##    term        estimate std.error statistic   p.value  nobs
##    <chr>          <dbl>     <dbl>     <dbl>     <dbl> <int>
##  1 (Intercept)     2.00    0.0633      31.6 1.66e-152  1000
##  2 X               4.87    0.110       44.4 1.69e-238  1000
##  3 (Intercept)     1.95    0.0614      31.8 1.16e-153  1000
##  4 X               5.21    0.106       49.1 3.22e-268  1000
##  5 (Intercept)     2.14    0.0632      33.8 8.12e-168  1000
##  6 X               4.32    0.109       39.5 3.55e-206  1000
##  7 (Intercept)     2.00    0.0592      33.9 4.17e-168  1000
##  8 X               4.95    0.102       48.4 8.43e-264  1000
##  9 (Intercept)     1.96    0.0627      31.3 2.82e-150  1000
## 10 X               5.19    0.108       47.8 2.62e-260  1000
```

```
print(summary(pool(lr_imp_MAR)))
```

```
##          term estimate std.error statistic       df      p.value
## 1 (Intercept) 2.011305 0.1035367  19.42601 9.455224 6.122136e-09
## 2           X 4.907146 0.4076786  12.03680 4.327239 1.724040e-04
```

```
dat_imp = complete(imp, "long", inc = TRUE)
```

After multiple imputation, the coefficients are accurate and standard errors are similar with the true values.

5

a

```
dat$Y_MNAR = ifelse(dat$Y > 5, NA, dat$Y)
```

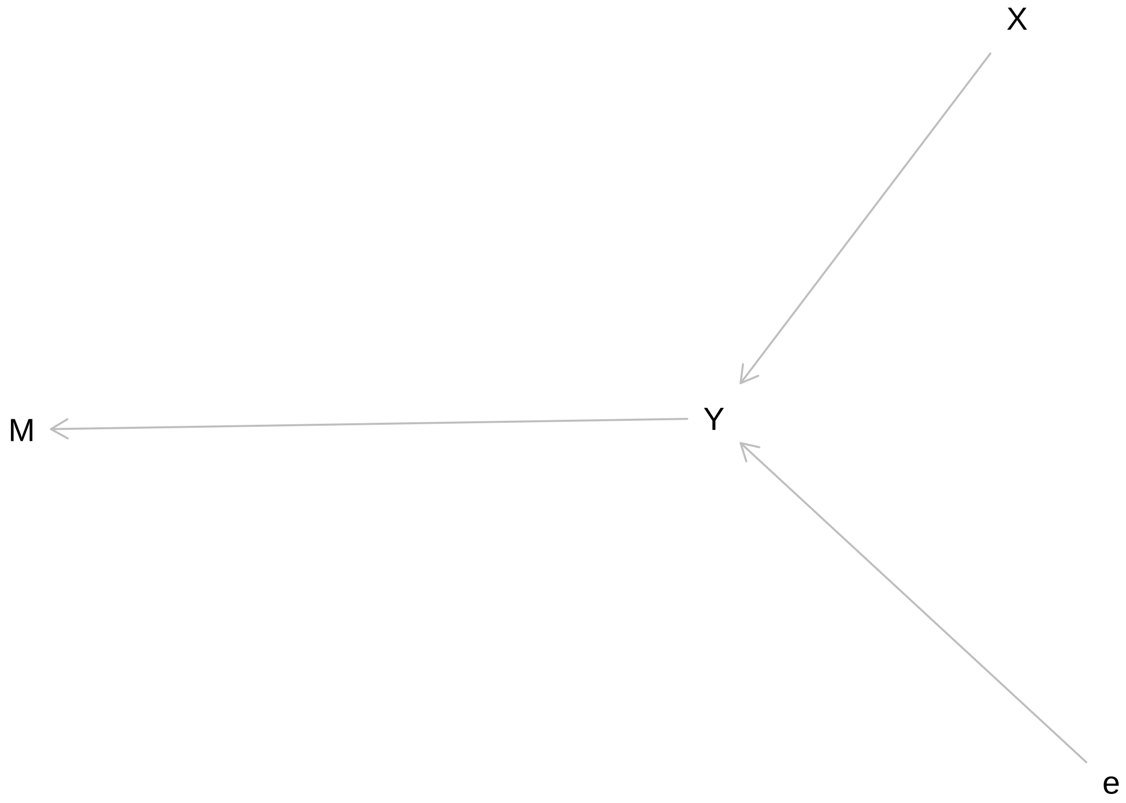
b

This is missing not at random mechanism. Because the missing of Y is dependent on Y.

c

```
set.seed(519)
g = dagitty('dag {
  X [exposure]
  Y [outcome]
  X -> Y
  e  -> Y
  Y -> M
  }')
plot(g)
```

```
## Plot coordinates for graph not supplied! Generating coordinates, see ?coordinates for how to set you
```



d

```
lr_MNAR = lm(Y_MNAR ~ X, data = dat)
print(summary(lr_MNAR))
```

```
##
## Call:
## lm(formula = Y_MNAR ~ X, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.97525 -0.53720  0.06106  0.57397  2.22358
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.28196    0.06338   36.00   <2e-16 ***
## X            3.13163    0.16092   19.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8494 on 594 degrees of freedom
##   (404 observations deleted due to missingness)
## Multiple R-squared:  0.3893, Adjusted R-squared:  0.3883
## F-statistic: 378.7 on 1 and 594 DF,  p-value: < 2.2e-16
```

The coefficients are not accurate and standard errors are larger.

e

```
imp = mice(dat[, c('X', 'Y_MNAR')], method="norm", m=5)
```

```
##
##  iter imp variable
##   1   1  Y_MNAR
##   1   2  Y_MNAR
##   1   3  Y_MNAR
##   1   4  Y_MNAR
##   1   5  Y_MNAR
##   2   1  Y_MNAR
##   2   2  Y_MNAR
##   2   3  Y_MNAR
##   2   4  Y_MNAR
##   2   5  Y_MNAR
##   3   1  Y_MNAR
##   3   2  Y_MNAR
##   3   3  Y_MNAR
##   3   4  Y_MNAR
##   3   5  Y_MNAR
##   4   1  Y_MNAR
##   4   2  Y_MNAR
##   4   3  Y_MNAR
##   4   4  Y_MNAR
##   4   5  Y_MNAR
##   5   1  Y_MNAR
##   5   2  Y_MNAR
##   5   3  Y_MNAR
##   5   4  Y_MNAR
##   5   5  Y_MNAR
```

```
lr_imp_MNAR = with(imp, lm(Y_MNAR ~ X))
print(summary(lr_imp_MNAR))
```

```
## # A tibble: 10 x 6
##    term        estimate std.error statistic   p.value  nobs
##    <chr>          <dbl>     <dbl>     <dbl>     <dbl> <int>
##  1 (Intercept)     2.35    0.0527      44.6 9.23e-240  1000
##  2 X               2.82    0.0913      30.9 5.88e-148  1000
##  3 (Intercept)     2.30    0.0557      41.3 2.70e-218  1000
##  4 X               3.04    0.0965      31.5 1.16e-151  1000
##  5 (Intercept)     2.35    0.0541      43.4 3.94e-232  1000
##  6 X               2.94    0.0936      31.5 1.97e-151  1000
##  7 (Intercept)     2.30    0.0546      42.1 1.50e-223  1000
##  8 X               3.09    0.0945      32.7 5.65e-160  1000
##  9 (Intercept)     2.27    0.0535      42.4 1.98e-225  1000
## 10 X               3.27    0.0925      35.4 3.34e-178  1000
```

```
print(summary(pool(lr_imp_MNAR)))
```

```
##          term estimate  std.error statistic        df      p.value
## 1 (Intercept) 2.313693 0.06685389  34.60821 32.088536 5.556959e-27
## 2           X 3.033608 0.20568999  14.74845  6.178427 4.766355e-06
```

```
dat_imp = complete(imp, "long", inc = TRUE)
```

After multiple imputation, the coefficients are still not accurate and standard errors are still larger, which means that the imputation is invalid in missing not at random mechanism.

6

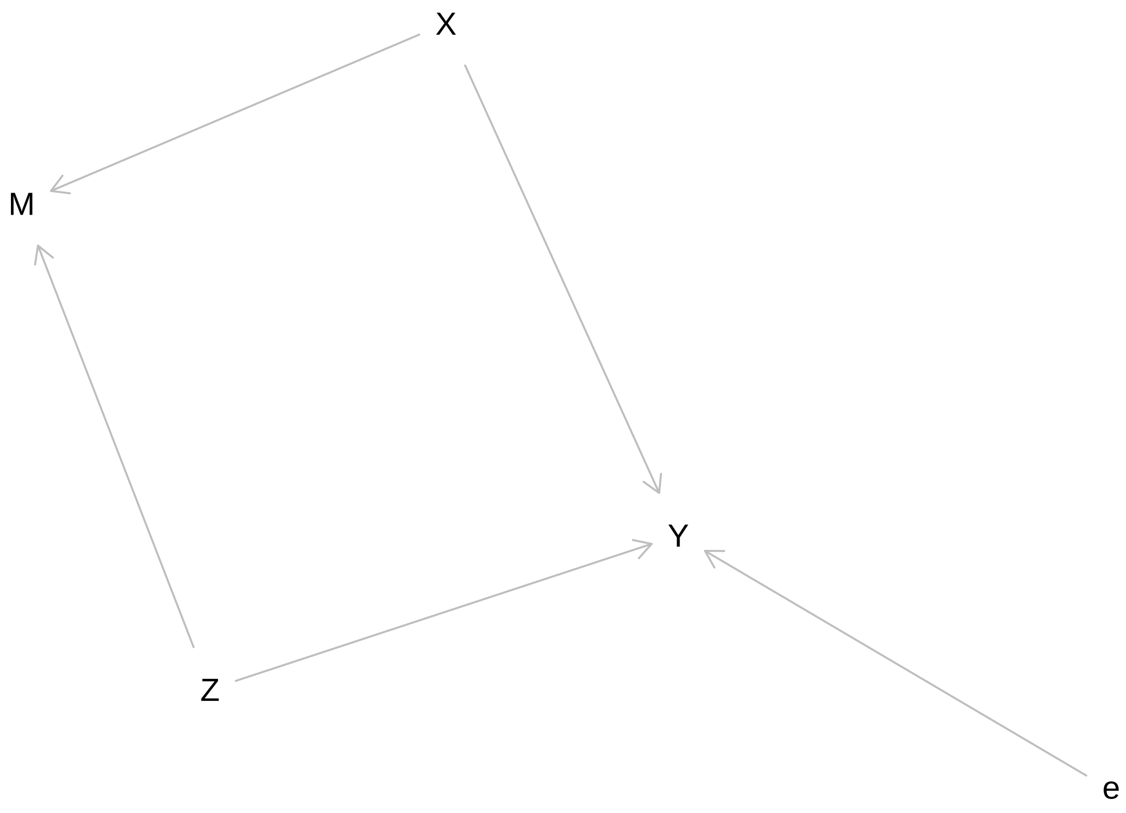
a

```
set.seed(19)
dat1 = dat[, c('X', 'e', 'Y')]
dat1$Z = rnorm(1000, mean = 0, sd = 1)
dat1$Y = 2 + 5*dat1$X + dat1$Z + dat1$e
```

b

```
dat1$Y_MAR = ifelse((dat1$X > 0.5) & (dat1$Z > 0), NA, dat1$Y)
set.seed(519)
g = dagitty('dag {
  X [exposure]
  Y [outcome]
  X -> {M Y}
  Z -> {M Y}
  e  -> Y
  }')
plot(g)
```

```
## Plot coordinates for graph not supplied! Generating coordinates, see ?coordinates for how to set you
```



c

```r
lr_MAR1 = lm(Y_MAR ~ X, data = dat1)
print(summary(lr_MAR1))
```

```
##
## Call:
## lm(formula = Y_MAR ~ X, data = dat1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6553 -0.8952  0.0083  0.8396  3.8499
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.17021    0.08804   24.65   <2e-16 ***
## X            4.03290    0.17620   22.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.332 on 750 degrees of freedom
##   (248 observations deleted due to missingness)
## Multiple R-squared:  0.4113, Adjusted R-squared:  0.4105
## F-statistic: 523.9 on 1 and 750 DF,  p-value: < 2.2e-16
```

The estimates of the parameters are not accurate.

d

```
imp = mice(dat1[, c('X', 'Y_MAR')], method="norm", m=5)
```

```
##
##  iter imp variable
##   1   1  Y_MAR
##   1   2  Y_MAR
##   1   3  Y_MAR
##   1   4  Y_MAR
##   1   5  Y_MAR
##   2   1  Y_MAR
##   2   2  Y_MAR
##   2   3  Y_MAR
##   2   4  Y_MAR
##   2   5  Y_MAR
##   3   1  Y_MAR
##   3   2  Y_MAR
##   3   3  Y_MAR
##   3   4  Y_MAR
##   3   5  Y_MAR
##   4   1  Y_MAR
##   4   2  Y_MAR
##   4   3  Y_MAR
##   4   4  Y_MAR
##   4   5  Y_MAR
##   5   1  Y_MAR
##   5   2  Y_MAR
##   5   3  Y_MAR
##   5   4  Y_MAR
##   5   5  Y_MAR
```

```
lr_imp_MAR1 = with(imp, lm(Y_MAR ~ X))
print(summary(lr_imp_MAR1))
```

```
## # A tibble: 10 x 6
##    term        estimate std.error statistic   p.value  nobs
##    <chr>          <dbl>     <dbl>     <dbl>     <dbl> <int>
##  1 (Intercept)     2.13    0.0836      25.5 5.93e-111  1000
##  2 X               4.14    0.145       28.6 4.51e-132  1000
##  3 (Intercept)     2.20    0.0880      25.0 1.69e-107  1000
##  4 X               3.98    0.152       26.1 3.80e-115  1000
##  5 (Intercept)     2.13    0.0851      25.1 7.02e-108  1000
##  6 X               4.25    0.147       28.8 2.08e-133  1000
##  7 (Intercept)     2.15    0.0840      25.6 2.49e-111  1000
##  8 X               4.10    0.145       28.2 2.69e-129  1000
##  9 (Intercept)     2.17    0.0837      25.9 8.40e-114  1000
## 10 X               4.06    0.145       28.1 3.88e-128  1000
```

```r
print(summary(pool(lr_imp_MAR1)))
```

```
##          term estimate  std.error statistic        df      p.value
## 1 (Intercept) 2.157154 0.09071274  23.78006 200.63607 2.761344e-60
## 2           X 4.107560 0.18164583  22.61302  31.87013 3.417377e-21
```

```r
dat_imp = complete(imp, "long", inc = TRUE)
```

After multiple imputation, the estimates of the parameters are still not accurate.

e

```r
lr_MAR1 = lm(Y_MAR ~ X + Z, data = dat1)
print(summary(lr_MAR1))
```

```
##
## Call:
## lm(formula = Y_MAR ~ X + Z, data = dat1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.01705 -0.66301 -0.01479  0.63171  2.93568
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.95401    0.06648   29.39   <2e-16 ***
## X            5.16996    0.13992   36.95   <2e-16 ***
## Z            0.98153    0.04040   24.29   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9971 on 749 degrees of freedom
##   (248 observations deleted due to missingness)
## Multiple R-squared:  0.6707, Adjusted R-squared:  0.6698
## F-statistic: 762.8 on 2 and 749 DF,  p-value: < 2.2e-16
```

```r
imp = mice(dat1[, c('X', 'Z', 'Y_MAR')], method="norm", m=5)
```

```
##
##  iter imp variable
##   1   1  Y_MAR
##   1   2  Y_MAR
##   1   3  Y_MAR
##   1   4  Y_MAR
##   1   5  Y_MAR
##   2   1  Y_MAR
##   2   2  Y_MAR
##   2   3  Y_MAR
##   2   4  Y_MAR
##   2   5  Y_MAR
##   3   1  Y_MAR
##   3   2  Y_MAR
##   3   3  Y_MAR
##   3   4  Y_MAR
##   3   5  Y_MAR
##   4   1  Y_MAR
##   4   2  Y_MAR
##   4   3  Y_MAR
##   4   4  Y_MAR
##   4   5  Y_MAR
##   5   1  Y_MAR
##   5   2  Y_MAR
##   5   3  Y_MAR
##   5   4  Y_MAR
##   5   5  Y_MAR
```

```
lr_imp_MAR1 = with(imp, lm(Y_MAR ~ X + Z))
print(summary(lr_imp_MAR1))
```

```
## # A tibble: 15 x 6
##    term        estimate std.error statistic   p.value  nobs
##    <chr>          <dbl>     <dbl>     <dbl>     <dbl> <int>
##  1 (Intercept)    1.95     0.0618      31.6 2.50e-152  1000
##  2 X              5.20     0.107       48.6 6.36e-265  1000
##  3 Z              1.00     0.0315      31.9 3.96e-154  1000
##  4 (Intercept)    1.94     0.0633      30.7 4.86e-146  1000
##  5 X              5.14     0.110       46.9 1.47e-254  1000
##  6 Z              0.970    0.0322      30.1 5.14e-142  1000
##  7 (Intercept)    1.96     0.0635      30.9 1.52e-147  1000
##  8 X              5.15     0.110       46.9 1.56e-254  1000
##  9 Z              0.983    0.0323      30.4 2.22e-144  1000
## 10 (Intercept)    1.95     0.0631      30.9 1.64e-147  1000
## 11 X              5.18     0.109       47.4 6.79e-258  1000
## 12 Z              0.971    0.0321      30.2 5.54e-143  1000
## 13 (Intercept)    1.92     0.0617      31.2 2.01e-149  1000
## 14 X              5.26     0.107       49.2 3.81e-269  1000
## 15 Z              1.01     0.0314      32.1 7.63e-156  1000
```

```
print(summary(pool(lr_imp_MAR1)))
```

```
##          term  estimate  std.error statistic        df       p.value
## 1 (Intercept) 1.9455969 0.06448194  30.17274 551.02167 8.426363e-119
## 2           X 5.1856750 0.12016410  43.15495 102.43438  1.592616e-67
## 3           Z 0.9870862 0.03753609  26.29699  48.61239  2.121897e-30
```

Now the estimates of parameters are accurate.

f

If Z was measured, this is missing at random mechanism because the missing of Y depend on X and Z. And if Z was not measured, this is missing not at random mechanism.

7

a

```
dat$X_MCAR = ifelse(dat$miss == 1, NA, dat$X)
lr_MCAR1 = lm(Y ~ X_MCAR, data = dat)
print(summary(lr_MCAR1))
```

```
##
## Call:
## lm(formula = Y ~ X_MCAR, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.96276 -0.66732 -0.01346  0.66040  2.84857
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.91275    0.09096   21.03   <2e-16 ***
## X_MCAR       5.11342    0.15603   32.77   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.007 on 502 degrees of freedom
##   (496 observations deleted due to missingness)
## Multiple R-squared:  0.6815, Adjusted R-squared:  0.6808
## F-statistic:  1074 on 1 and 502 DF,  p-value: < 2.2e-16
```

The estimates of parameters are accurate, but standard errors are larger.

b

```
setup = mice(dat[, c('X_MCAR', 'Y')])
```

```
##
##  iter imp variable
##   1   1  X_MCAR
##   1   2  X_MCAR
##   1   3  X_MCAR
##   1   4  X_MCAR
##   1   5  X_MCAR
##   2   1  X_MCAR
##   2   2  X_MCAR
##   2   3  X_MCAR
##   2   4  X_MCAR
##   2   5  X_MCAR
##   3   1  X_MCAR
##   3   2  X_MCAR
##   3   3  X_MCAR
##   3   4  X_MCAR
##   3   5  X_MCAR
##   4   1  X_MCAR
##   4   2  X_MCAR
##   4   3  X_MCAR
##   4   4  X_MCAR
##   4   5  X_MCAR
##   5   1  X_MCAR
##   5   2  X_MCAR
##   5   3  X_MCAR
##   5   4  X_MCAR
##   5   5  X_MCAR
```

```
predMat = setup$predictorMatrix
predMat["X_MCAR","Y"] = 0
imp = mice(dat[, c('X_MCAR', 'Y')], method="norm", predictorMatrix=predMat, m = 5)
```

```
##
##  iter imp variable
##   1   1  X_MCAR
##   1   2  X_MCAR
##   1   3  X_MCAR
##   1   4  X_MCAR
##   1   5  X_MCAR
##   2   1  X_MCAR
##   2   2  X_MCAR
##   2   3  X_MCAR
##   2   4  X_MCAR
##   2   5  X_MCAR
##   3   1  X_MCAR
##   3   2  X_MCAR
##   3   3  X_MCAR
##   3   4  X_MCAR
##   3   5  X_MCAR
##   4   1  X_MCAR
##   4   2  X_MCAR
##   4   3  X_MCAR
##   4   4  X_MCAR
##   4   5  X_MCAR
##   5   1  X_MCAR
##   5   2  X_MCAR
##   5   3  X_MCAR
##   5   4  X_MCAR
##   5   5  X_MCAR
```

c

```
lr_imp_MCAR1 = with(imp, lm(Y ~ X_MCAR))
print(summary(lr_imp_MCAR1))
```

```
## # A tibble: 10 x 6
##    term        estimate std.error statistic   p.value  nobs
##    <chr>          <dbl>    <dbl>    <dbl>     <dbl> <int>
##  1 (Intercept)     3.31    0.103     32.1 4.76e-156  1000
##  2 X_MCAR          2.41    0.176     13.6 5.16e- 39  1000
##  3 (Intercept)     3.11    0.107     29.0 1.07e-134  1000
##  4 X_MCAR          2.73    0.182     15.0 5.07e- 46  1000
##  5 (Intercept)     3.29    0.104     31.7 2.09e-153  1000
##  6 X_MCAR          2.40    0.176     13.7 4.50e- 39  1000
##  7 (Intercept)     3.20    0.105     30.5 8.75e-145  1000
##  8 X_MCAR          2.55    0.178     14.4 1.26e- 42  1000
##  9 (Intercept)     3.16    0.107     29.6 1.87e-138  1000
## 10 X_MCAR          2.64    0.183     14.5 4.01e- 43  1000
```

```
print(summary(pool(lr_imp_MCAR1)))
```

```
##          term estimate std.error statistic       df      p.value
## 1 (Intercept) 3.214656 0.1403251  22.90863 20.09300 7.129734e-16
## 2      X_MCAR 2.545020 0.2370708  10.73527 20.76963 6.236353e-10
```

```
dat_imp = complete(imp, "long", inc = TRUE)
```

The estimates are not accurate.

d

```
imp = mice(dat[, c('X_MCAR', 'Y')], method="norm", m = 5)
```

```
##
##  iter imp variable
##   1   1  X_MCAR
##   1   2  X_MCAR
##   1   3  X_MCAR
##   1   4  X_MCAR
##   1   5  X_MCAR
##   2   1  X_MCAR
##   2   2  X_MCAR
##   2   3  X_MCAR
##   2   4  X_MCAR
##   2   5  X_MCAR
##   3   1  X_MCAR
##   3   2  X_MCAR
##   3   3  X_MCAR
##   3   4  X_MCAR
##   3   5  X_MCAR
##   4   1  X_MCAR
##   4   2  X_MCAR
##   4   3  X_MCAR
##   4   4  X_MCAR
##   4   5  X_MCAR
##   5   1  X_MCAR
##   5   2  X_MCAR
##   5   3  X_MCAR
##   5   4  X_MCAR
##   5   5  X_MCAR
```

```
lr_imp_MCAR1 = with(imp, lm(Y ~ X_MCAR))
print(summary(lr_imp_MCAR1))
```

```
## # A tibble: 10 x 6
##    term        estimate std.error statistic   p.value  nobs
##    <chr>          <dbl>     <dbl>     <dbl>     <dbl> <int>
##  1 (Intercept)     1.80    0.0653      27.5 1.76e-124  1000
##  2 X_MCAR          5.35    0.112       47.6 1.01e-258  1000
##  3 (Intercept)     1.90    0.0647      29.4 1.78e-137  1000
##  4 X_MCAR          5.11    0.110       46.4 1.10e-251  1000
##  5 (Intercept)     1.90    0.0638      29.8 6.29e-140  1000
##  6 X_MCAR          5.13    0.109       47.2 3.36e-256  1000
##  7 (Intercept)     1.99    0.0621      32.1 8.25e-156  1000
##  8 X_MCAR          5.02    0.106       47.3 7.11e-257  1000
##  9 (Intercept)     1.96    0.0648      30.2 4.74e-143  1000
## 10 X_MCAR          5.09    0.112       45.6 2.34e-246  1000
```

```
print(summary(pool(lr_imp_MCAR1)))
```

```
##          term estimate std.error statistic       df      p.value
## 1 (Intercept) 1.909735 0.1034766  18.45571 10.26245 3.313881e-09
## 2      X_MCAR 5.140013 0.1738496  29.56587 10.78301 1.141638e-11
```

```
dat_imp = complete(imp, "long", inc = TRUE)
```

After imputing X on Y, the parameters become accurate.

Part B

1

a

```
dat = read.csv('necosad_death_miss.csv', stringsAsFactors = TRUE)[, -c(1, 2)]
```

b

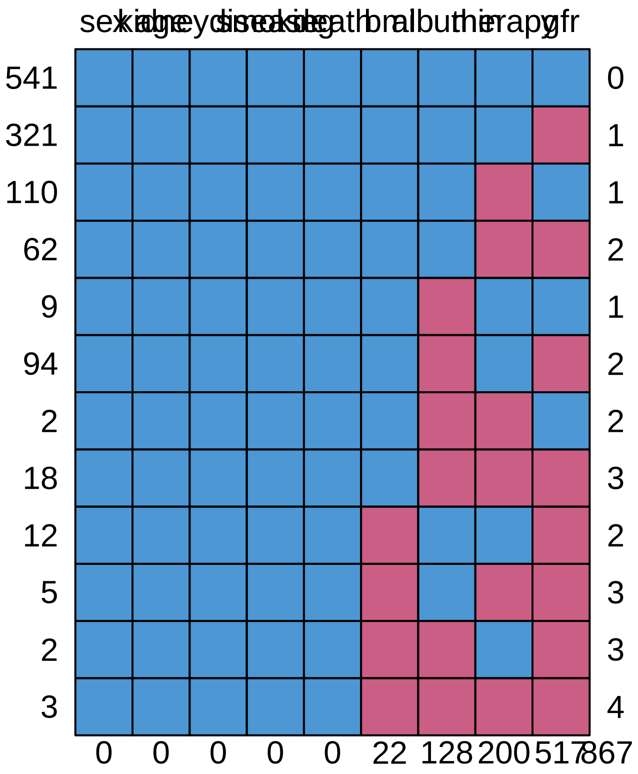
```
str(dat)
```

```
## 'data.frame':    1179 obs. of  9 variables:
##  $ sex          : Factor w/ 2 levels "female","male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ therapy      : Factor w/ 2 levels "hemodialysis",..: 1 2 1 1 NA 2 NA 1 2 2 ...
##  $ age          : num  54.9 47.2 53.9 46.1 54.3 62.5 56 49.2 55 54.2 ...
##  $ bmi          : num  40.4 29.6 28.7 27.7 21.9 ...
##  $ albumin      : num  45 34 NA 42 34 21.6 40 23 39.9 45 ...
##  $ kidneydisease: Factor w/ 4 levels "Diabetes Mellitus",..: 3 4 3 1 2 4 4 2 3 4 ...
##  $ smoking      : Factor w/ 2 levels "current or former smoker",..: 1 2 1 1 1 1 1 1 1 1 ...
##  $ gfr          : num  NA 5.68 NA 3.57 4.39 ...
##  $ death        : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
```

```
summary(dat)
```

```
##      sex                     therapy          age             bmi
##  female:458   hemodialysis       :691   Min.   :18.50   Min.   : 2.044
##  male  :721   peritoneal dialysis:288   1st Qu.:58.40   1st Qu.:22.070
##               NA's               :200   Median :68.50   Median :24.435
##                                         Mean   :65.49   Mean   :25.220
##                                         3rd Qu.:74.70   3rd Qu.:27.432
##                                         Max.   :91.60   Max.   :94.451
##                                                         NA's   :22
##     albumin                    kidneydisease                     smoking
##  Min.   : 9.00   Diabetes Mellitus     :210   current or former smoker:852
##  1st Qu.:31.30   Glomerulonephritis    : 89   never smoker            :327
##  Median :35.60   Renal Vascular Disease:269
##  Mean   :35.02   other                 :611
##  3rd Qu.:39.00
##  Max.   :67.00
##  NA's   :128
##       gfr         death
##  Min.   : 0.000   no :390
##  1st Qu.: 2.686   yes:789
##  Median : 4.617
##  Mean   : 5.069
##  3rd Qu.: 6.995
##  Max.   :51.742
##  NA's   :517
```

```
md.pattern(dat)
```



```
##     sex age kidneydisease smoking death bmi albumin therapy gfr    
## 541   1   1             1       1     1   1       1       1   1   0
## 321   1   1             1       1     1   1       1       1   0   1
## 110   1   1             1       1     1   1       1       0   1   1
## 62    1   1             1       1     1   1       1       0   0   2
## 9     1   1             1       1     1   1       0       1   1   1
## 94    1   1             1       1     1   1       0       1   0   2
## 2     1   1             1       1     1   1       0       0   1   2
## 18    1   1             1       1     1   1       0       0   0   3
## 12    1   1             1       1     1   0       1       1   0   2
## 5     1   1             1       1     1   0       1       0   0   3
## 2     1   1             1       1     1   0       0       1   0   3
## 3     1   1             1       1     1   0       0       0   0   4
##        0   0             0       0     0  22     128     200 517 867
```

2

a

```
log_r = glm(death ~ therapy, data = dat, family = 'binomial')
print(summary(log_r))
```

```
##
```

```
## Call:
## glm(formula = death ~ therapy, family = "binomial", data = dat)
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.83561    0.08282  10.089  < 2e-16 ***
## therapyperitoneal dialysis -0.41275    0.14621  -2.823  0.00476 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1241.6  on 978  degrees of freedom
## Residual deviance: 1233.7  on 977  degrees of freedom
##   (200 observations deleted due to missingness)
## AIC: 1237.7
##
## Number of Fisher Scoring iterations: 4
```

```
log_r1 = glm(death ~ therapy + sex + age + bmi + albumin + kidneydisease + smoking + gfr, data = dat, fa
sum1 = summary(log_r1)
print(sum1)
```

```
##
## Call:
## glm(formula = death ~ therapy + sex + age + bmi + albumin + kidneydisease +
##     smoking + gfr, family = "binomial", data = dat)
##
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                          1.534316   1.099558   1.395  0.16290
## therapyperitoneal dialysis           0.306623   0.232479   1.319  0.18719
## sexmale                             -0.343947   0.235324  -1.462  0.14385
## age                                  0.053959   0.009136   5.906 3.50e-09 ***
## bmi                                 -0.049839   0.024303  -2.051  0.04029 *
## albumin                             -0.045755   0.018620  -2.457  0.01400 *
## kidneydiseaseGlomerulonephritis     -1.963485   0.439271  -4.470 7.83e-06 ***
## kidneydiseaseRenal Vascular Disease -0.290167   0.388872  -0.746  0.45556
## kidneydiseaseother                  -1.473419   0.317978  -4.634 3.59e-06 ***
## smokingnever smoker                 -0.647499   0.241810  -2.678  0.00741 **
## gfr                                  0.009339   0.028687   0.326  0.74476
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 671.92  on 540  degrees of freedom
## Residual deviance: 557.75  on 530  degrees of freedom
##   (638 observations deleted due to missingness)
## AIC: 579.75
##
## Number of Fisher Scoring iterations: 4
```

b?

```
print(c(model1=exp(log_r$coefficients[2]), model2=exp(log_r1$coefficients[2])))
```

```
## model1.therapyperitoneal dialysis model2.therapyperitoneal dialysis
##                         0.6618257                         1.3588289
```

The odds ratio for therapy in model without covariates is 0.66 and in model with covariates is 1.36. It means that in model without covariates the risk of death in PD group is 0.66 times that of HD group, in model with covariates the risk of death in PD group is 1.36 times higher than HD group.

c

979 patients in model without covariates and 541 in model with covariates. The sample size changes, and we cannot make sure the difference in odds ratio is due to adding confounders or modeling on different subset of samples.

3

a

```
set.seed(519)
ini = mice(dat, maxit = 0)
meth = ini$meth
meth["gfr"] = "norm"
imp = mice(dat, method = meth, m = 5)
```

```
##
##  iter imp variable
##   1   1  therapy  bmi  albumin  gfr
##   1   2  therapy  bmi  albumin  gfr
##   1   3  therapy  bmi  albumin  gfr
##   1   4  therapy  bmi  albumin  gfr
##   1   5  therapy  bmi  albumin  gfr
##   2   1  therapy  bmi  albumin  gfr
##   2   2  therapy  bmi  albumin  gfr
##   2   3  therapy  bmi  albumin  gfr
##   2   4  therapy  bmi  albumin  gfr
##   2   5  therapy  bmi  albumin  gfr
##   3   1  therapy  bmi  albumin  gfr
##   3   2  therapy  bmi  albumin  gfr
##   3   3  therapy  bmi  albumin  gfr
##   3   4  therapy  bmi  albumin  gfr
##   3   5  therapy  bmi  albumin  gfr
##   4   1  therapy  bmi  albumin  gfr
##   4   2  therapy  bmi  albumin  gfr
##   4   3  therapy  bmi  albumin  gfr
##   4   4  therapy  bmi  albumin  gfr
##   4   5  therapy  bmi  albumin  gfr
##   5   1  therapy  bmi  albumin  gfr
##   5   2  therapy  bmi  albumin  gfr
##   5   3  therapy  bmi  albumin  gfr
##   5   4  therapy  bmi  albumin  gfr
##   5   5  therapy  bmi  albumin  gfr
```

```
dat_imp = complete(imp, "long", inc = TRUE)
```

b

```
log_r1_imp = with(imp, glm(death ~ therapy + sex + age + bmi + albumin + kidneydisease + smoking + gfr,
print(summary(log_r1_imp))
```

```
## # A tibble: 55 x 6
##    term                              estimate std.error statistic  p.value  nobs
##    <chr>                                <dbl>     <dbl>     <dbl>    <dbl> <int>
##  1 (Intercept)                         0.545     0.665       0.820 4.12e- 1  1179
##  2 therapyperitoneal dialysis          0.119     0.154       0.773 4.39e- 1  1179
##  3 sexmale                           -0.116     0.148      -0.785 4.33e- 1  1179
##  4 age                                 0.0520    0.00584     8.90  5.71e-19  1179
##  5 bmi                               -0.00992   0.0136     -0.732 4.64e- 1  1179
##  6 albumin                           -0.0508    0.0117     -4.33  1.51e- 5  1179
##  7 kidneydiseaseGlomerulonephritis   -1.55      0.301      -5.15  2.57e- 7  1179
##  8 kidneydiseaseRenal Vascular Dise~ -0.474     0.256      -1.85  6.37e- 2  1179
##  9 kidneydiseaseother                -1.32      0.214      -6.13  8.64e-10  1179
## 10 smokingnever smoker               -0.284     0.160      -1.77  7.64e- 2  1179
## # i 45 more rows
```

```
pool_sum1 = summary(pool(log_r1_imp))
print(pool_sum1)
```

```
##                                    term     estimate   std.error  statistic
## 1                           (Intercept)  0.784541378 0.690849243  1.1356188
## 2            therapyperitoneal dialysis  0.113615978 0.163819033  0.6935457
## 3                               sexmale -0.122409329 0.148455082 -0.8245547
## 4                                   age  0.051574834 0.005883233  8.7664099
## 5                                   bmi -0.009202543 0.013743063 -0.6696137
## 6                               albumin -0.056382733 0.013179561 -4.2780434
## 7       kidneydiseaseGlomerulonephritis -1.559353992 0.303325666 -5.1408574
## 8   kidneydiseaseRenal Vascular Disease -0.454043861 0.257339190 -1.7643790
## 9                    kidneydiseaseother -1.317694514 0.215189823 -6.1234054
## 10                  smokingnever smoker -0.288540600 0.162069015 -1.7803564
## 11                                  gfr -0.021068115 0.024202982 -0.8704760
##            df      p.value
## 1   514.75844 2.566446e-01
## 2   336.79373 4.884452e-01
## 3  1158.84110 4.097942e-01
## 4  1135.98111 6.610771e-18
## 5  1005.55836 5.032578e-01
## 6    95.31190 4.483076e-05
## 7  1155.07920 3.209568e-07
## 8  1071.78155 7.795300e-02
## 9  1151.71478 1.254556e-09
## 10 1086.00381 7.529717e-02
## 11   22.73283 3.931398e-01
```

```
print(exp(sum1$coefficients[, 1]))
```

```
##                        (Intercept)          therapyperitoneal dialysis
##                          4.6381497                           1.3588289
##                            sexmale                                 age
##                          0.7089662                           1.0554410
##                                bmi                             albumin
##                          0.9513827                           0.9552764
##     kidneydiseaseGlomerulonephritis kidneydiseaseRenal Vascular Disease
##                          0.1403683                           0.7481384
##                 kidneydiseaseother                  smokingnever smoker
##                          0.2291406                           0.5233530
##                                gfr
##                          1.0093832
```

```
print(exp(pool_sum1$estimate))
```

```
##  [1] 2.1914017 1.1203218 0.8847861 1.0529280 0.9908397 0.9451773 0.2102719
##  [8] 0.6350549 0.2677519 0.7493564 0.9791523
```

The odds ratio for therapy after imputation is 1.12, which is 1.36 before imputation. The value changed but not much.

c

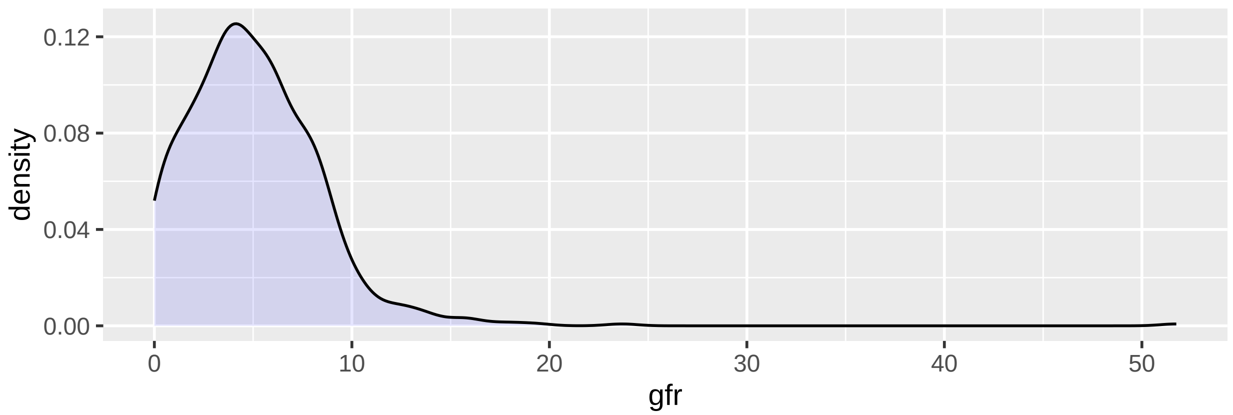
```
library(ggplot2)
```

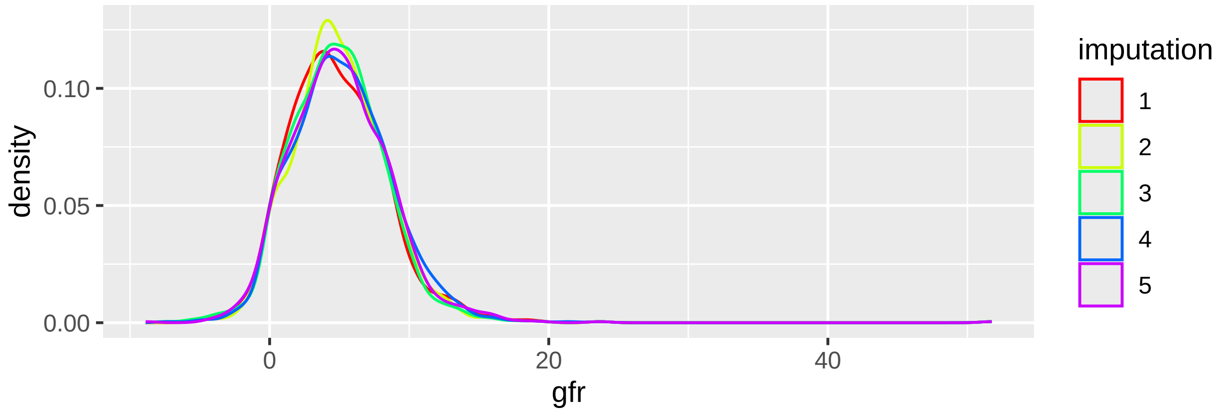
```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
library(gridExtra)
```

```
p1 = ggplot(data = dat, mapping = aes(x = gfr)) +
    geom_density(fill = 'blue', alpha = 0.1)
p2 = ggplot(data = dat_imp[dat_imp$.imp != 0, ], mapping = aes(x = gfr, group = as.factor(.imp), colour
    geom_density() +
    scale_color_manual(values = rainbow(5)) +
    guides(color = guide_legend(title = 'imputation'))
grid.arrange(p1, p2, nrow = 2)
```

```
## Warning: Removed 517 rows containing non-finite outside the scale range
## (`stat_density()`).
```





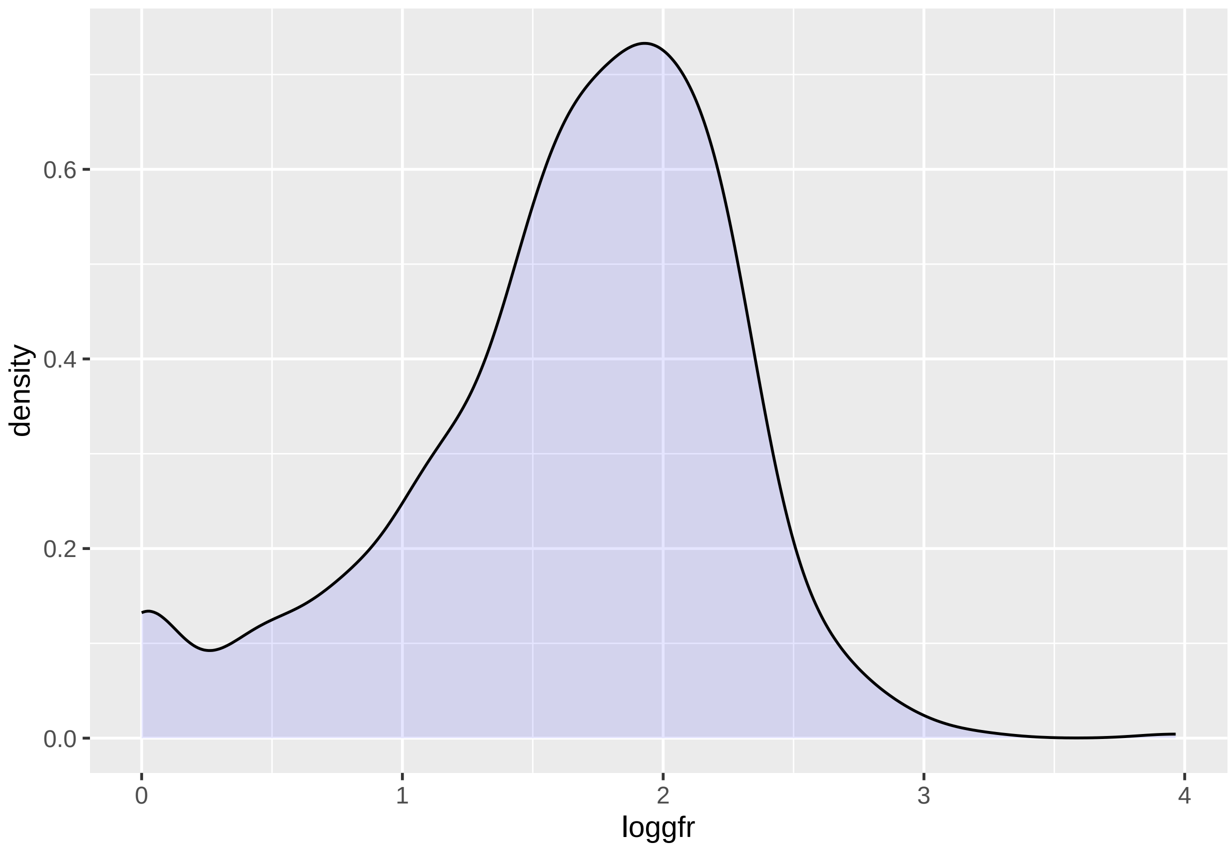
No, imputation applied a symmetric distribution to generate gfr and generated negative gfr values.

4

a

```
dat$loggfr = log(dat$gfr+1)
ggplot(data = dat, mapping = aes(x = loggfr)) +
    geom_density(fill = 'blue', alpha = 0.1)
```

```
## Warning: Removed 517 rows containing non-finite outside the scale range
## (‘stat_density()‘).
```



b

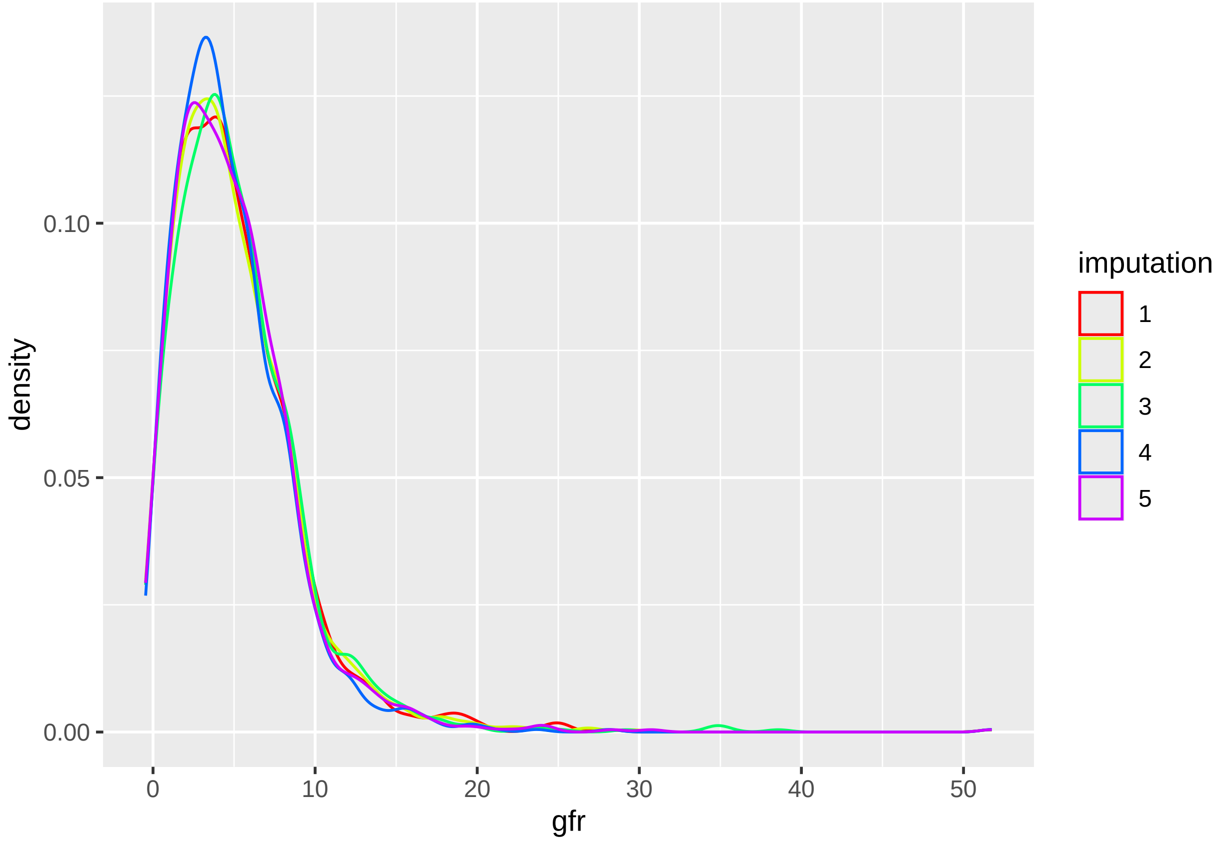
```
ini = mice(dat, maxit = 0)
# define methods for imputation
meth["loggfr"] = "norm"
meth["gfr"] = "~I(exp(loggfr)-1)"
# and do not use gfr in the imputation models
predMat = ini$predictorMatrix
predMat[,"gfr"] = 0
imp = mice(dat, method = meth, predictorMatrix = predMat, m = 5)
```

```
##
##  iter imp variable
##   1   1  therapy  bmi  albumin  gfr  loggfr
##   1   2  therapy  bmi  albumin  gfr  loggfr
##   1   3  therapy  bmi  albumin  gfr  loggfr
##   1   4  therapy  bmi  albumin  gfr  loggfr
##   1   5  therapy  bmi  albumin  gfr  loggfr
##   2   1  therapy  bmi  albumin  gfr  loggfr
##   2   2  therapy  bmi  albumin  gfr  loggfr
##   2   3  therapy  bmi  albumin  gfr  loggfr
##   2   4  therapy  bmi  albumin  gfr  loggfr
##   2   5  therapy  bmi  albumin  gfr  loggfr
##   3   1  therapy  bmi  albumin  gfr  loggfr
##   3   2  therapy  bmi  albumin  gfr  loggfr
##   3   3  therapy  bmi  albumin  gfr  loggfr
##   3   4  therapy  bmi  albumin  gfr  loggfr
##   3   5  therapy  bmi  albumin  gfr  loggfr
##   4   1  therapy  bmi  albumin  gfr  loggfr
##   4   2  therapy  bmi  albumin  gfr  loggfr
##   4   3  therapy  bmi  albumin  gfr  loggfr
##   4   4  therapy  bmi  albumin  gfr  loggfr
##   4   5  therapy  bmi  albumin  gfr  loggfr
##   5   1  therapy  bmi  albumin  gfr  loggfr
##   5   2  therapy  bmi  albumin  gfr  loggfr
##   5   3  therapy  bmi  albumin  gfr  loggfr
##   5   4  therapy  bmi  albumin  gfr  loggfr
##   5   5  therapy  bmi  albumin  gfr  loggfr
```

```
dat_imp = complete(imp, "long", inc = TRUE)
```

c

```
ggplot(data = dat_imp[dat_imp$.imp != 0, ], mapping = aes(x = gfr, group = as.factor(.imp), colour = as
    geom_density() +
    scale_color_manual(values = rainbow(5)) +
    guides(color = guide_legend(title = 'imputation'))
```



The imputed gfr values now follow an asymmetric distribution, but still have negative values. It's more plausible.

d

```r
log_r1_imp1 = with(imp, glm(death ~ therapy + sex + age + bmi + albumin + kidneydisease + smoking + gfr
print(summary(log_r1_imp1))
```

```
## # A tibble: 55 x 6
##    term                              estimate std.error statistic  p.value  nobs
##    <chr>                                <dbl>     <dbl>     <dbl>    <dbl> <int>
##  1 (Intercept)                         0.803    0.673       1.19  2.33e- 1  1179
##  2 therapyperitoneal dialysis          0.228    0.161       1.42  1.56e- 1  1179
##  3 sexmale                            -0.131    0.148      -0.881 3.78e- 1  1179
##  4 age                                 0.0523   0.00588     8.89  5.87e-19  1179
##  5 bmi                                -0.0114   0.0135     -0.846 3.97e- 1  1179
##  6 albumin                            -0.0535   0.0122     -4.38  1.20e- 5  1179
##  7 kidneydiseaseGlomerulonephritis    -1.62     0.303      -5.33  9.67e- 8  1179
##  8 kidneydiseaseRenal Vascular Dise~  -0.493    0.257      -1.92  5.52e- 2  1179
##  9 kidneydiseaseother                 -1.34     0.216      -6.23  4.74e-10  1179
## 10 smokingnever smoker                -0.328    0.162      -2.03  4.29e- 2  1179
## # i 45 more rows
```

```r
pool_sum2 = summary(pool(log_r1_imp1))
print(pool_sum2)
```

```
##                                    term     estimate   std.error  statistic
## 1                           (Intercept)  0.810146591 0.690243750  1.1737109
## 2            therapyperitoneal dialysis  0.175539803 0.175472670  1.0003826
## 3                               sexmale -0.125247946 0.148365828 -0.8441832
## 4                                   age  0.051902688 0.005924041  8.7613657
## 5                                   bmi -0.009785105 0.013700673 -0.7142062
## 6                               albumin -0.055290926 0.012957365 -4.2671427
## 7       kidneydiseaseGlomerulonephritis -1.593356827 0.304759164 -5.2282491
## 8   kidneydiseaseRenal Vascular Disease -0.471934781 0.257660991 -1.8316113
## 9                    kidneydiseaseother -1.329328725 0.216973567 -6.1266851
## 10                  smokingnever smoker -0.310417402 0.161693766 -1.9197858
## 11                                  gfr -0.033953732 0.019032203 -1.7840148
##             df      p.value
## 1    687.35820 2.409174e-01
## 2    107.51584 3.193716e-01
## 3   1161.52445 3.987408e-01
## 4   1047.98893 7.634496e-18
## 5   1090.61997 4.752525e-01
## 6    176.62623 3.223515e-05
## 7   1129.98955 2.037412e-07
## 8   1110.82107 6.727703e-02
## 9   1092.33692 1.250560e-09
## 10  1147.01384 5.513277e-02
## 11    99.53027 7.746868e-02
```

```
print(exp(sum1$coefficients[, 1]))
```

```
##                          (Intercept)          therapyperitoneal dialysis
##                            4.6381497                           1.3588289
##                              sexmale                                 age
##                            0.7089662                           1.0554410
##                                  bmi                             albumin
##                            0.9513827                           0.9552764
##     kidneydiseaseGlomerulonephritis kidneydiseaseRenal Vascular Disease
##                            0.1403683                           0.7481384
##                   kidneydiseaseother                 smokingnever smoker
##                            0.2291406                           0.5233530
##                                  gfr
##                            1.0093832
```

```
print(exp(pool_sum2$estimate))
```

```
##  [1] 2.2482375 1.1918894 0.8822781 1.0532732 0.9902626 0.9462098 0.2032422
##  [8] 0.6237942 0.2646549 0.7331409 0.9666162
```

The odds ratio for therapy after new imputation is 1.14, which is 1.36 in b and 1.12 in c. The value improved but not much. Because the imputation of other variables have not been checked.