*Assignment*1

Shukang Guo Github: SueGK ID: s2040164
URL: https://github.com/SueGK/IDA_Assignment1

1.

(a) Answer:(ii) 0.3

The probability of ALQ being missing for those with $ALQ = No$ is 0.3, which equals to the ones for those with $ALQ = Yes$. Because ALQ is MCAR, the missing and observed values will have similar distributions.

(b) Answer:(ii)

The probability of ALQ being missing is independent of the Yes/No value of ALQ after adjusting for gender and only depends on the value of gender(which is the observed/available information).Because MAR assumption implies that the probability that a ALQ value is missing varies with the gender but does not depend on the ALQ themselves.

(b) Answer:(iii)

It is impossible to conclude from the information given. There is no information about the probability of ALQ being missing for women. The probability for men and women are independent.

2.

Under a complete case analysis, it excludes the data for any case/individual that has one or more missing values. The largest possible subsample is a sample consists of 90 subjects and 10 variables where the missing values of all 10 variable are from the same 10 subjects which would be discarded. The smallest subsample is a sample consists of 0 subjects and 0 variables where the missing values of all 10 variable are from the different 10 subjects. So all subjects are discarded, we only got a zero subsample.

3.

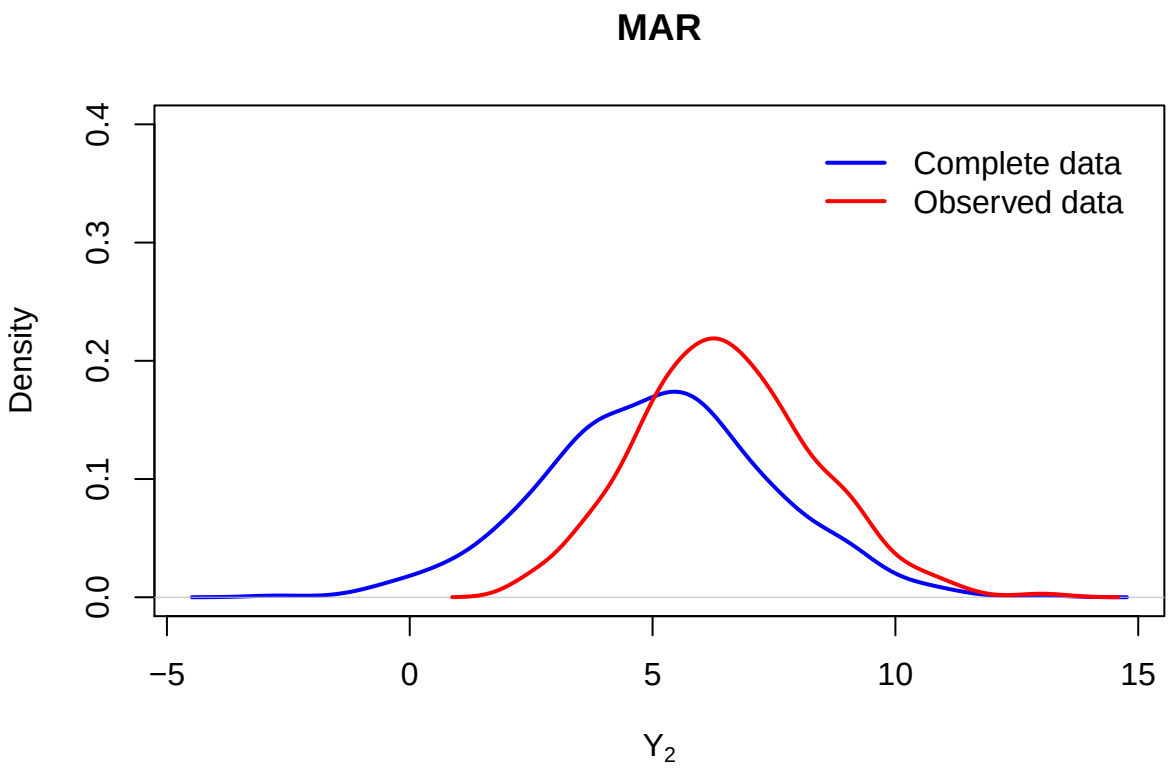
```
require(MASS)
#Simulating the data
n = 500
```

```r
set.seed(1)
z1 = rnorm(n,0,1)
set.seed(2)
z2 = rnorm(n,0,1)
set.seed(3)
z3 = rnorm(n,0,1)

y1 = 1 + z1
y2 = 5+2*z1+z2
data_y = data.frame(y1, y2)
```

(a) Answer: MAR

Y_2 is missing if $2 \times (Y_1 - 1) + Z_3 < 0$: The mechanism is MAR. Since it depends on Y_1(fully observed). According to the figure below, We can observe that under the MAR mechanism, the two distributions are different.

```r
ind_y2 = which(2*(y1-1)+z3 < 0)
y2_mis = y2[ind_y2]
y2_obs = y2[-ind_y2]

#Display the marginal distribution of y2
plot(density(y2), lwd = 2, col = "blue", xlab = expression(Y[2]), main= "MAR",
     ylim = c(0, 0.4))

lines(density(y2_obs), lwd = 2, col = "red")

legend(8, 0.4, legend = c("Complete data", "Observed data"),
       col = c("blue", "red"), lty = c(1,1), lwd = c(2,2), bty ="n")
```



(b)

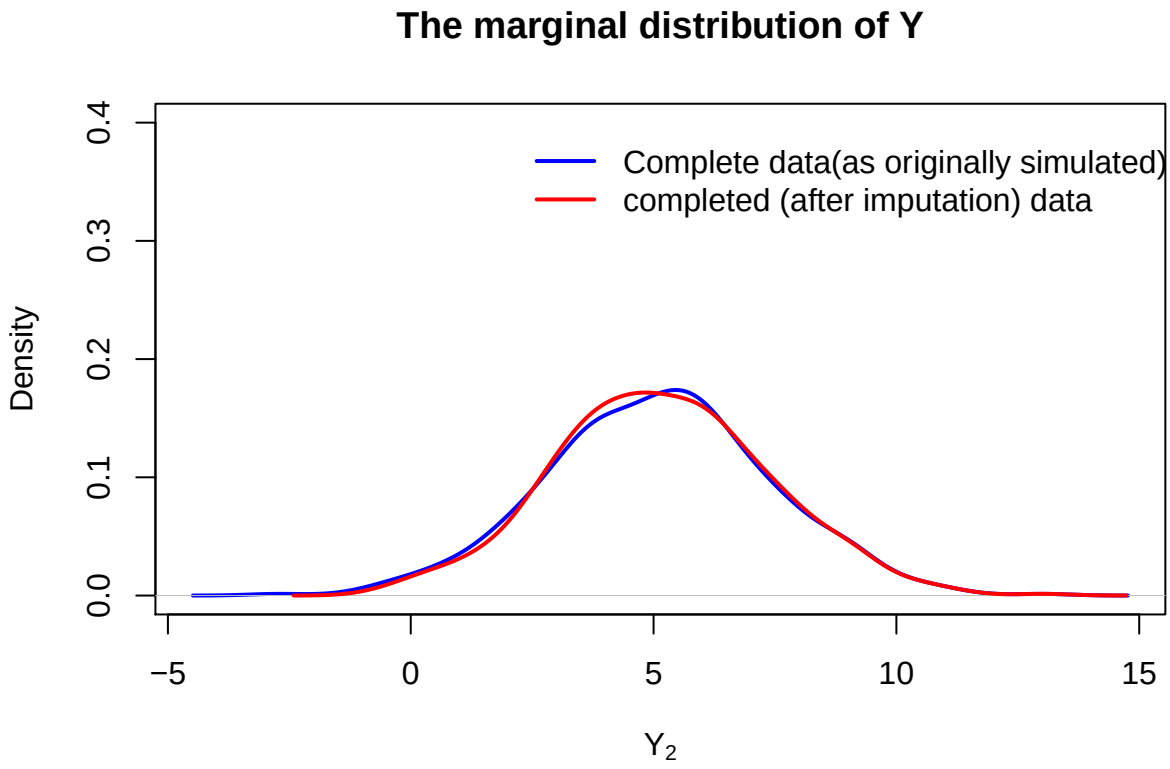
We will fit a linear regression model to the complete cases, using y_2 as the response and Y_1 as the predictors, that is,

$$\widehat{Y_2} = \widehat{\beta}_0 + \widehat{\beta}_1 Y_1 + z, \quad z \sim \mathrm{N}\left(0, \widehat{\sigma}^2\right), \quad \widehat{\sigma} = 1.027704$$

As we can observe, the marginal distribution of Y2 for the complete (as originally simulated) and completed (after imputation) data are same. So we can think it fits the model perfectly by imputing the missing values using stochastic regression imputation.

```
#y2_new converts missing values in y2 to NA
y2_new=ifelse(2*(y1-1)+z3 < 0, NA, data_y$y2)
data_y = data.frame(y1, y2_new)
fitt = lm(y2_new  ~ y1, data = data_y)
set.seed(4)
#replace the NA with predicted values
pred <- predict(fitt, newdata = data_y) + rnorm(n, 0, summary(fitt)$sigma)
sri_y2 <- ifelse(is.na(data_y$y2_new) == TRUE, pred, data_y$y2_new)

#Display the marginal distribution of Y2 for the complete (as originally
#simulated) and completed (after imputation) data
plot(density(y2), lwd = 2, col = "blue", xlab = expression(Y[2]),
     main= "The marginal distribution of Y",
     ylim = c(0, 0.4))
```

```r
lines(density(sri_y2), lwd = 2, col = "red")

legend(2, 0.4, legend = c("Complete data(as originally simulated)",
                          "completed (after imputation) data"),
       col = c("blue", "red"), lty = c(1,1), lwd = c(2,2), bty ="n")
```

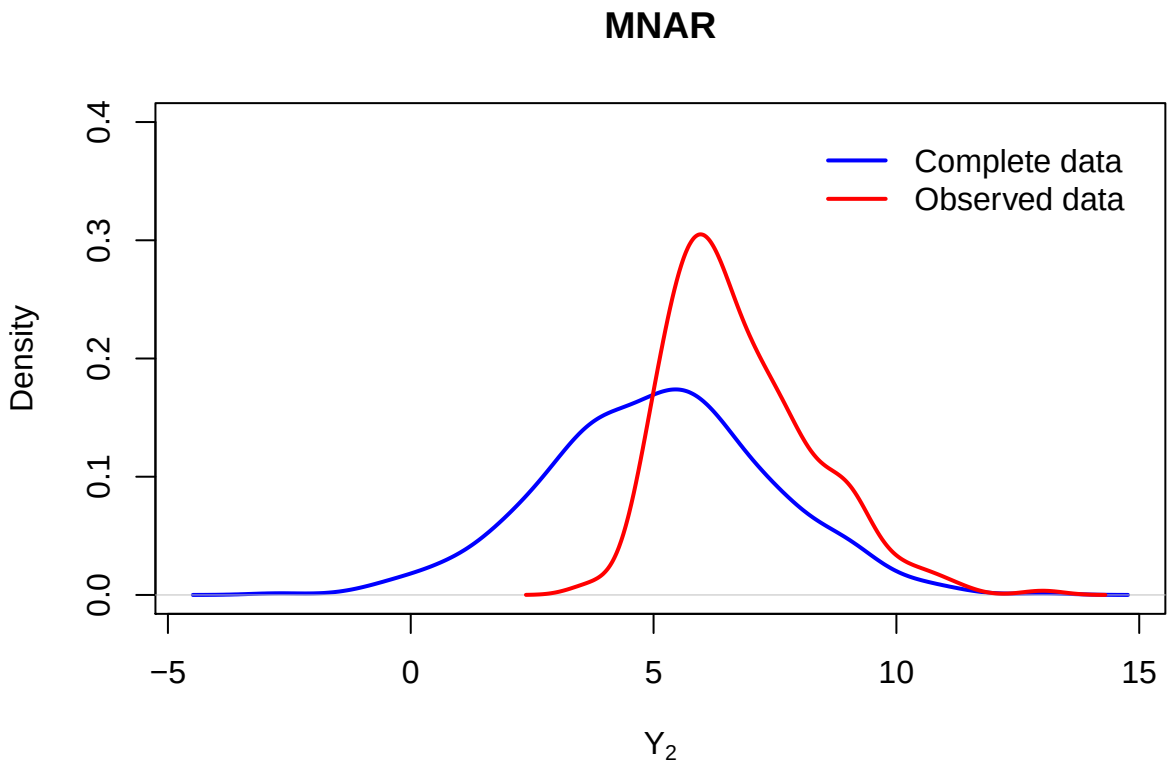


(c) Answer: MNAR

Y_2 is missing if $2 \times (Y_2 - 5) + Z_3 < 0$: The mechanism is MNAR. Since it depends on both Y_2(which has missing values) and Z_3(the specific values that should have been obtained). According to the figure below, We can observe that under the MNAR mechanism, the MNAR case is even more extreme in terms of the dissimilarities between the two distributions.

```r
newind_y2 = which(2*(y2-5)+z3 < 0)
newy2_mis = y2[newind_y2]
newy2_obs = y2[-newind_y2]

#Display the marginal distribution of y2
plot(density(y2), lwd = 2, col = "blue", xlab = expression(Y[2]), main= "MNAR",
     ylim = c(0, 0.4))

lines(density(newy2_obs), lwd = 2, col = "red")
```

```
legend(8, 0.4, legend = c("Complete data", "Observed data"),
       col = c("blue", "red"), lty = c(1,1), lwd = c(2,2), bty ="n")
```



(d)

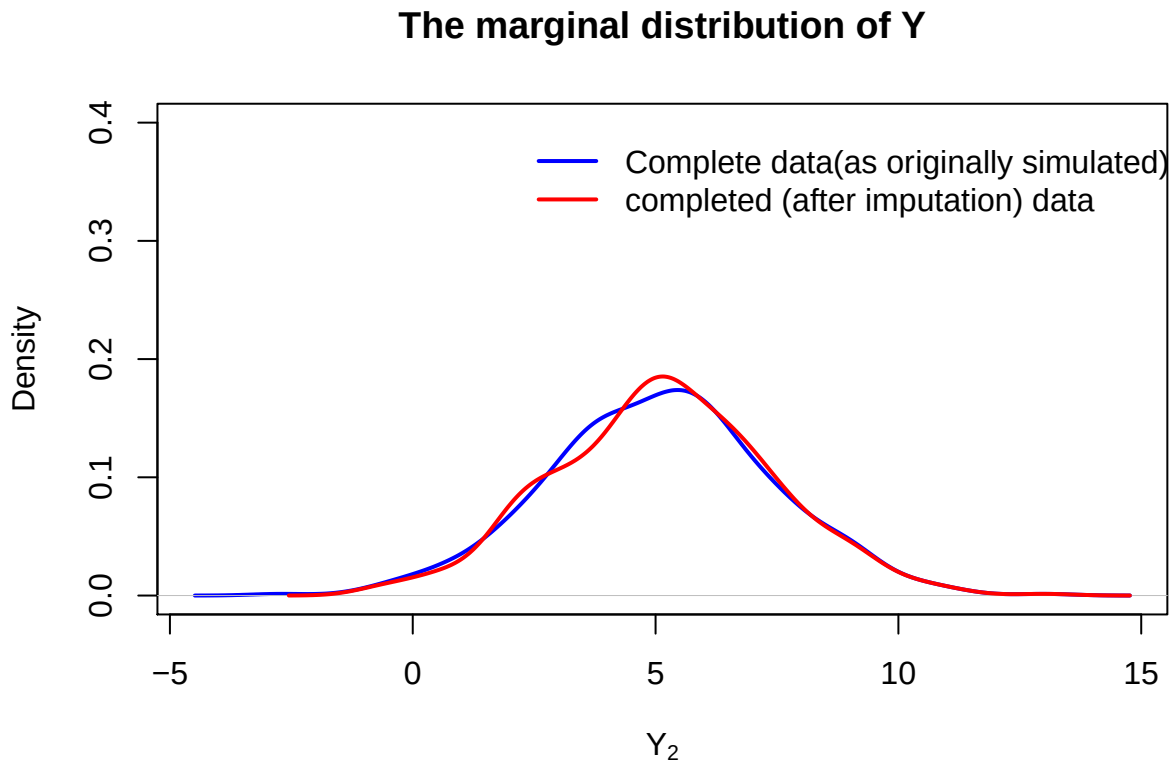
We will fit a linear regression model to the complete cases, using y_2 as the response and Y_1 as the predictors, that is,

$$\widehat{Y_2} = \widehat{\beta}_0 + \widehat{\beta}_1 Y_1 + z, \quad z \sim \mathrm{N}\left(0, \widehat{\sigma}^2\right), \quad \widehat{\sigma} = 1.027704$$

As we can observe, the marginal distribution of Y2 for the complete (as originally simulated) and completed (after imputation) data are same. So we can think it fits the model perfectly.

```
#y2_new converts missing values in y2 to NA
y2_new=ifelse(2*(y2-5)+z3 < 0, NA, data_y$y2)
fitt = lm(y2_new  ~ y1, data = data_y)
set.seed(5)
#replace the NA with predicted values
pred <- predict(fitt, newdata = data_y) + rnorm(n, 0, summary(fitt)$sigma)
sri_y2 <- ifelse(is.na(data_y$y2_new) == TRUE, pred, data_y$y2_new)

#Display the marginal distribution of Y2 f
plot(density(y2), lwd = 2, col = "blue", xlab = expression(Y[2]),
     main= "The marginal distribution of Y",
     ylim = c(0, 0.4))
```

```
lines(density(sri_y2), lwd = 2, col = "red")

legend(2, 0.4, legend = c("Complete data(as originally simulated)",
                          "completed (after imputation) data"),
       col = c("blue", "red"), lty = c(1,1), lwd = c(2,2), bty ="n")
```



4. Stochastic regression imputation

(a) Complete case analysis

```
load("databp.Rdata")
```

The mean value of the recovery time is 19.27273 and associated standard error is 2.603013. The (Pearson) correlations between the recovery time and the dose is 0.2391256, and between the recovery time and blood pressure is -0.01952862.

```
# mean value of the recovery time and associated standard error
ind = which(is.na(databp$recovtime) == FALSE)
mrecov = mean(databp$recovtime, na.rm = TRUE)
serecov = sd(databp$recovtime, na.rm = TRUE)/sqrt(length(ind))
mrecov;serecov
```

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

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

```
#the (Pearson) correlations between the recovery time and the dose
correcov_dose = cor(databp$recovtime,databp$logdose,
                    use = "complete",
                    method = "pearson")

#the (Pearson) correlations between the recovery time and blood pressure
correcov_blood = cor(databp$recovtime,databp$bloodp,
                     use = "complete",
                     method = "pearson")

correcov_dose;correcov_blood
```

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

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

(b) Using mean imputation

The mean recovery time using mean imputation is 19.27273. This value is equal to the one obtained in the overall complete case analysis (after mean imputation, the mean of the dataset remains unchanged). The associated standard error is 2.284135, which is smaller than the one provided by an overall complete case analysis (in mean imputation, each imputed value contributes with a zero value to the numerator, but the denominator is inflated, when compared to the complete cases analysis).

The (Pearson) correlations between the recovery time and the dose, and between the recovery time and blood pressure are 0.2150612 and -0.01934126 respectively, which are smaller than the one provided by an overall complete case analysis (in the covariance formula. Cases with missing values on either one of the variables contribute with a value of zero to the numerator formula).

```
mirecov = ifelse(is.na(databp$recovtime) == TRUE,
                 mean(databp$recovtime,na.rm = TRUE), databp$recovtime)

# mean value of the recovery time and associated standard error
n = length(mirecov)
mimrecov = mean(mirecov)
miserecov = sd(mirecov)/sqrt(n)
mimrecov;miserecov
```

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

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

```
#the (Pearson) correlations between the recovery time and the dose
mi_correcov_dose = cor(mirecov,databp$logdose, method = "pearson")
#the (Pearson) correlations between the recovery time and blood pressure
mi_correcov_blood = cor(mirecov,databp$bloodp, method = "pearson")

mi_correcov_dose;mi_correcov_blood
```

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

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

(c) Using mean regression imputation

We will fit a linear regression model to the complete cases, using vitamin D as the response and dose and blood pressure as the predictors, that is,

$$recovtime = \beta_0 + \beta_1 logdose + \beta_2 bloodp + \varepsilon, \varepsilon \sim \mathrm{N}\left(0, \sigma^2\right)$$

The estimated regression coefficients are $\beta_0 = 15.2159$, $\beta_1 = 11.4290$ and $\beta_2 = -0.2769$. Then we should replace the NA with predicted values. The mean recovery time using mean imputation is 19.44428. The associated standard error is 2.312845. The (Pearson) correlations between the recovery time and the dose, and between the recovery time and blood pressure are 0.2801835 and -0.0111364 respectively.

```
fitrecov = lm(recovtime ~ logdose + bloodp, data = databp)
#summary(fitrecov)

#replace the NA with predicted values
predri <- predict(fitrecov,newdata=databp)
#predri[4]; predri[10]; predri[22]

mrirecov = ifelse(is.na(databp$recovtime) == TRUE, predri, databp$recovtime)

# mean value of the recovery time and associated standard error
n = length(mrirecov)
mrimrecov = mean(mrirecov)
mriserecov = sd(mrirecov)/sqrt(n)
mrimrecov;mriserecov
```

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

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

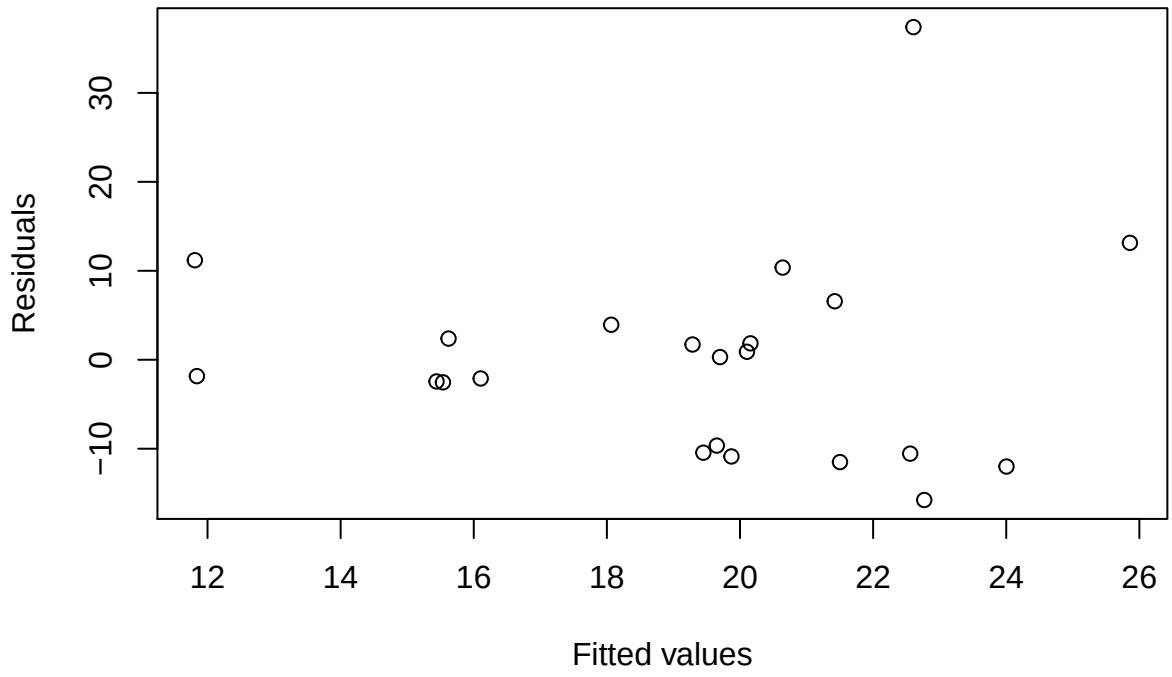
```
#the (Pearson) correlations between the recovery time and the dose
mri_correcov_dose = cor(mrirecov,databp$logdose, method = "pearson")
#the (Pearson) correlations between the recovery time and blood pressure
mri_correcov_blood = cor(mrirecov,databp$bloodp, method = "pearson")

mri_correcov_dose;mri_correcov_blood
```

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

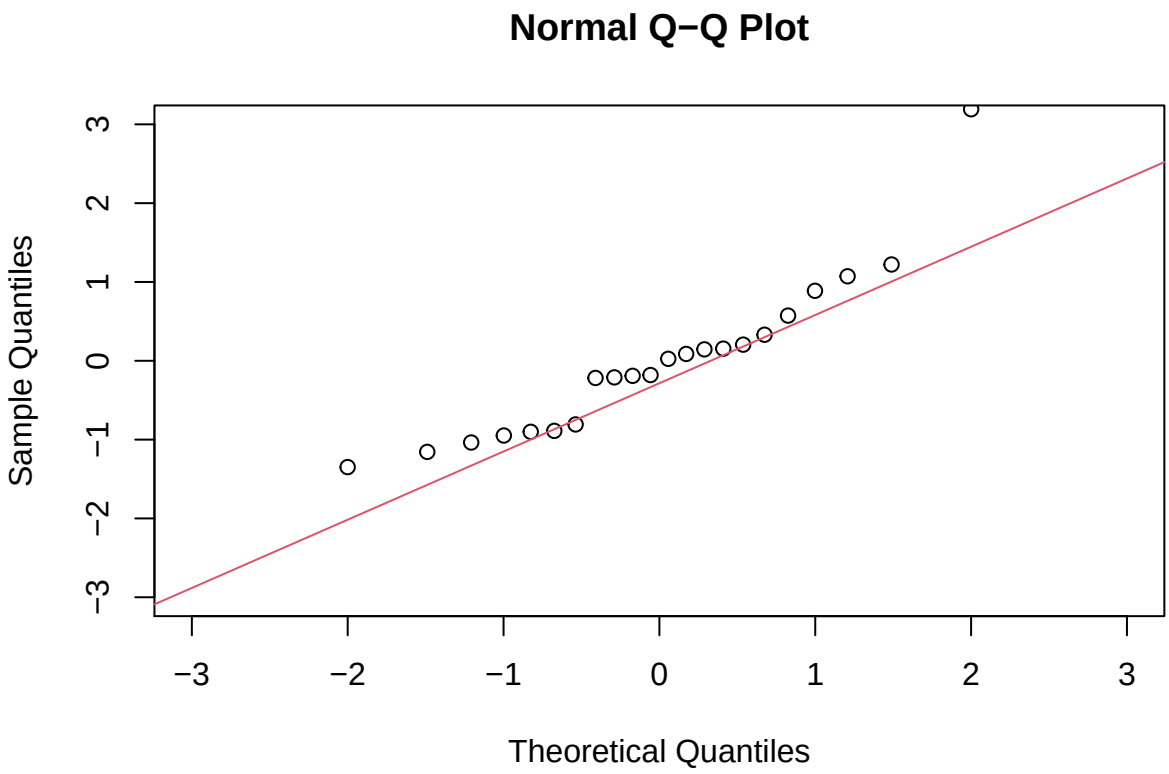
```
## [1] -0.0111364
```

Then we neek to check the validity of model's assumptions by plotting of the residuals to check the assumption of linearity (and homoscedasticity). According to the figure below, there is no obvious pattern, so there is no reason to suspect of a nonlinear relationship or of no constant variance. We can also check the assumption of normality of error terms using a QQ-plot.

```r
plot(fitrecov$fitted.values, residuals(fitrecov), xlab = "Fitted values", ylab = "Residuals")
```




```r
#QQ-Plot
qqnorm(rstandard(fitrecov), xlim = c(-3,3), ylim = c(-3,3))
qqline(rstandard(fitrecov),col=2)
```



(d) Using stochastic regression imputation

We can fit a similar linear regression model to the one in (c), but add a random noise.

$$\widehat{\text{recovtime}} = \widehat{\beta}_0 + \widehat{\beta}_1 logdose + \widehat{\beta}_2 bloodp + z, \quad z \sim \text{N}\left(0, \widehat{\sigma}^2\right), \quad \widehat{\sigma} = 12.25$$

Then we should replace the NA with predicted values. The mean recovery time using stochastic regression imputation is 20.4598. The associated standard error is 2.444571. The (Pearson) correlations between the recovery time and the dose, and between the recovery time and blood pressure are 0.2284537 and -0.01786944 respectively.

When conducting stochastic regression imputation, we should know that in some cases, stochastic regression imputation, by adding a random noise term, can lead to implausible predictions like imputing a negative value.

```
set.seed(1)
predsri <- predict(fitrecov, newdata = databp) + rnorm(n, 0, summary(fitrecov)$sigma)
#predsri[4]; predsri[10]; predsri[22]
recovsri <- ifelse(is.na(databp$recovtime) == TRUE, predsri, databp$recovtime)
msri <- mean(recovsri)
sesri <- sd(recovsri)/sqrt(n)
msri; sesri
```

```
## [1] 20.4598
```

```
## [1] 2.444571
```

```r
#the (Pearson) correlations between the recovery time and the dose
sri_correcov_dose = cor(recovsri,databp$logdose, method = "pearson")
#the (Pearson) correlations between the recovery time and blood pressure
sri_correcov_blood = cor(recovsri,databp$bloodp, method = "pearson")

sri_correcov_dose;sri_correcov_blood
```

```
## [1] 0.2284537
```

```
## [1] -0.01786944
```

(e) Predictive mean matching

The mean recovery time using predictive mean matching is 19.64051. The associated standard error is 2.300533. The (Pearson) correlations between the recovery time and the dose, and between the recovery time and blood pressure are 0.2801835 and -0.0111364 respectively.

```r
donor4 = c()
donor10 = c()
donor22 = c()

for (i in predri[-4]){
  sq =(i-predri[4])**2
  donor4=append(donor4, sq)
}
index1 = which.min(donor4)
donor1 = predri[index1]

for (i in predri[-10]){
  sq =(i-predri[10])**2
  donor10=append(donor10, sq)
}
index2 =which.min(donor10)
donor2 = predri[index2]

for (i in predri[-22]){
  sq =(i-predri[22])**2
  donor22=append(donor22, sq)
}
index3 =which.min(donor22)
donor3 = predri[index3]

#new recovtime
pmrecov <- c(databp$recovtime[is.na(databp$recovtime) == FALSE], donor1, donor2, donor3)
m=length(pmrecov)
mpm <- mean(pmrecov);
corpm <- sd(pmrecov)/sqrt(m)
mpm;corpm
```

```
## [1] 19.64051
```

```
## [1] 2.300533
```

```r
#the (Pearson) correlations between the recovery time and the dose
pm_correcov_dose = cor(mrirecov,databp$logdose, method = "pearson")
#the (Pearson) correlations between the recovery time and blood pressure
pm_correcov_blood = cor(mrirecov,databp$bloodp, method = "pearson")
pm_correcov_dose;pm_correcov_blood
```

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

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

(f) Predictive mean matching

The advantage of predictive mean matching does not attenuate the variability of the imputed data and impute values that are much more similar to real values. But the problem is that the prediction can not be good if we only little prediction data. In addition, if we got too much NA in the original dataset, the prediction for NA by predictive mean matching may be not robust.