IDA Assignment 1

Aijia Huang s2036061

## 1.

### (a) (ii) 0.3

### The MCAR means the missing data distribution and observed data distribution and complete data distribution are similar. Further, MCAR implies that the probability of missing value for the variable is the same for all individuals. And 0.3 in question means into ALQ=YES, the probability of missing value is 0.3. Therefore, ALQ=No is the same distribution with ALQ=Yes, and the answer should be 0.3.

### (b) (ii) The probability of ALQ being missing is independent of the Yes/No value of ALQ after adjusting for gender.

### Since ALQ is MAR, hence the missing values in ALQ dosen’t depend on themselves, and they should depend on other observed variable and data. So, (i) and (iii) are both incorrect.

### (c) (iii) It is impossible to conclude from the information given.

### The similar idea with question (a), however, MAR doesn’t have the same distribution of missing data, observed data and complete data. Also, the probability of missing value for the variable is not the same for all individuals. Therefore, although we know the probability of ALQ being missing for men is 0.1, it is impossible to conclude the question from that.

## 2.The largest possible subsample is 90, and the smallest subsample is 0.

### The dataset consists of 100 subjects and 10 variables. Each variable contains 10% of missing values (10 missing values). If 10% missing values of each variable are the same position, there will be the largest possible subsample, 100*(1-10%) = 90. If 10% missing values of each variable are the different and completely unique position, there will be the smallest possible subsample, 100*(1-10\*10%) = 0.

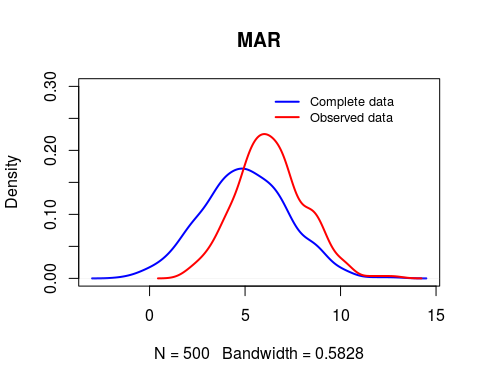
## 3.

### (a) MAR

### MAR:Since a=2 and b=0,hence in the missing condition: “a*(Y1-1)+b*(Y2-5)+Z3<0”, there isn’t any contributions of Y2. So the missing values of Y2 are unrelate to themselves and relate to Y1. Therefore, MAR.

### Comment: Under the MAR, as shown in the graph, the peak of the observed (after imposing missingness) data density moves right and become higher than the complete data, and the density of complete data and observed (after imposing missingness) data are not similar. This is consistent with the rule of MAR mechanism, that complete data and observed data distribution are different.

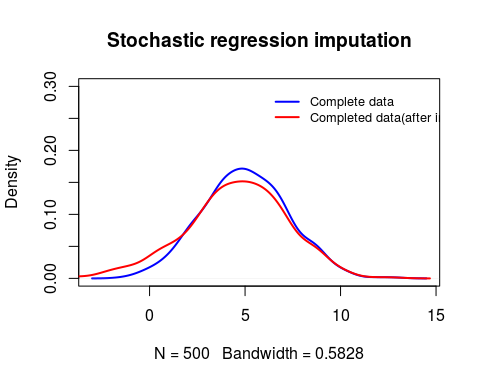
#Question 3  
# (a)  
set.seed(1)  
n<- 500  
mu1<-mu2<-mu3<-0  
sigma1<-sigma2<-sigma3<-1  
z1<-rnorm(n,mu1,sigma1)  
z2<-rnorm(n,mu2,sigma2)  
z3<-rnorm(n,mu3,sigma3)  
  
Y1<-1+z1  
Y2<-5+2\*z1+z2  
  
#MAR  
Y2\_MAR\_obs <- Y2[2\*z1 +z3 >= 0]  
Y2\_MAR\_mis <- Y2[2\*z1 +z3<0]  
  
plot(density(Y2), lwd = 2, col = "blue", main = "MAR", ylim = c(0,0.3))  
lines(density(Y2\_MAR\_obs), lwd = 2, col = "red")  
legend(6, 0.3,legend = c("Complete data", "Observed data"),cex=0.8,  
 col = c("blue", "red"), lty = c(1, 1), lwd = c(2, 2), bty = "n")



## (b)

### Comment: Imputing the missing values using stochastic regression imputation means here we use a regression model to predict the incomplete variables(Y2) from the complete variable(Y1), and then add noise to the predictions, which can restore lost variability to the data. Further, stochastic regression imputation can give unbiased parameter estimates under MAR. Hence, as shown in this graph, the density of complete data and data after imputation are similar compared with the graph in the Question(a). This also can show that the stochastic regression imputation approach is a great way to fill the missing data. And it can cause relatively less errors within this approach.

#Question 3  
# (b)  
set.seed(1)  
n<- 500  
mu1<-mu2<-mu3<-0  
sigma1<-sigma2<-sigma3<-1  
z1<-rnorm(n,mu1,sigma1)  
z2<-rnorm(n,mu2,sigma2)  
z3<-rnorm(n,mu3,sigma3)  
  
Y1<-1+z1  
Y2<-5+2\*z1+z2  
  
Y2c=Y2  
Y2c[which((2\*z1 +z3)<0)]=NA  
  
data<-data.frame(Y1=Y1, Y2cc=Y2c)  
  
fit <- lm(Y2cc ~ Y1, data = data)   
  
set.seed(1)  
predicted\_sri <- predict(fit, newdata = data) + rnorm(nrow(data), 0, sigma(fit))  
Y2\_sri <- ifelse(is.na(data$Y2cc) == TRUE, predicted\_sri, data$Y2cc)  
  
plot(density(Y2), lwd = 2, col = "blue", main = "Stochastic regression imputation", ylim = c(0,0.3))  
lines(density(Y2\_sri), lwd = 2, col = "red")  
legend(6, 0.3,legend = c("Complete data", "Completed data(after imputation)"),cex=0.8,  
 col = c("blue", "red"), lty = c(1, 1), lwd = c(2, 2), bty = "n")

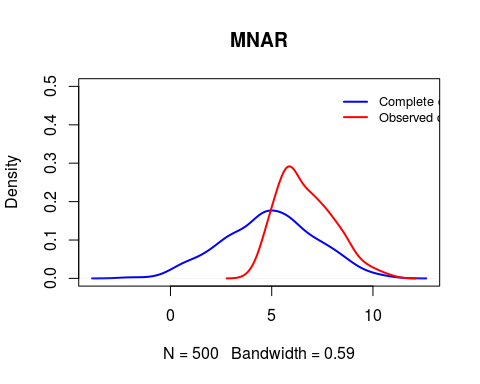


### (c) MNAR

### MNAR:Since a=0 and b=2, hence in the missing condition: “a*(Y1-1)+b*(Y2-5)+Z3<0”, there isn’t any contributions of Y1. So the missing values of Y2 are unrelate to Y1 and relate to Y2. Therefore, MNAR.

### Comment: Under the MNAR, as shown in the graph, the peak of the observed (after imposing missingness) data density moves right and become higher than the complete data, and the density of complete data and observed (after imposing missingness) data are quite different. This is consistent with the rule of MNAR mechanism, that complete data and observed data distribution are different.

#Question 3  
# (c)  
n<- 500  
mu1<-mu2<-mu3<-0  
sigma1<-sigma2<-sigma3<-1  
z1<-rnorm(n,mu1,sigma1)  
z2<-rnorm(n,mu2,sigma2)  
z3<-rnorm(n,mu3,sigma3)  
  
Y1<-1+z1  
Y2<-5+2\*z1+z2  
  
#MNAR  
Y2\_MNAR\_obs <- Y2[2\*Y2-10+z3 >= 0]  
Y2\_MNAR\_mis <- Y2[2\*Y2-10+z3 < 0]  
  
plot(density(Y2), lwd = 2, col = "blue", main = "MNAR", ylim = c(0,0.5))  
lines(density(Y2\_MNAR\_obs), lwd = 2, col = "red")  
legend(8, 0.5,legend = c("Complete data", "Observed data"),cex=0.8,  
 col = c("blue", "red"), lty = c(1, 1), lwd = c(2, 2), bty = "n")

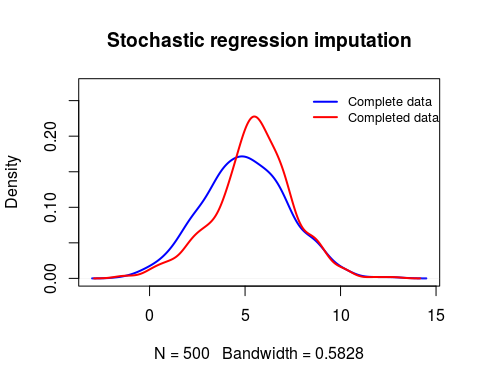


## Question 3

## (d)

### Comment: Under imputing the missing values using stochastic regression imputation, as shown in this graph, the density of complete data and data after imputation are relatively similar compared with graph in the Question(c). That implies the stochastic regression imputation is a great approach to fill the missing data. However, due to MNAR mechanism, they are still a bit different from the density of complete data and completed data after imputation.

#Question 3  
# (d)  
set.seed(1)  
n<- 500  
mu1<-mu2<-mu3<-0  
sigma1<-sigma2<-sigma3<-1  
z1<-rnorm(n,mu1,sigma1)  
z2<-rnorm(n,mu2,sigma2)  
z3<-rnorm(n,mu3,sigma3)  
  
Y1<-1+z1  
Y2<-5+2\*z1+z2  
  
Y2d=Y2  
Y2d[(4\*z1+2\*z2+z3)<0]=NA  
  
data\_d<-data.frame(Y1=c(Y1), Y2dd=c(Y2d))  
  
fit\_d<- lm(Y2dd ~ Y1, data = data\_d)   
  
set.seed(1)  
predicted\_sri\_d <- predict(fit\_d, newdata = data\_d) + rnorm(nrow(data\_d), 0, sigma(fit\_d))  
Y2\_sri\_d <- ifelse(is.na(data\_d$Y2dd), predicted\_sri\_d, data\_d$Y2dd)  
  
plot(density(Y2), lwd = 2, col = "blue", main = "Stochastic regression imputation", ylim = c(0,0.27))  
lines(density(Y2\_sri\_d), lwd = 2, col = "red")  
legend(8, 0.27,legend = c("Complete data", "Completed data(after imputation)"),cex=0.8,  
 col = c("blue", "red"), lty = c(1, 1), lwd = c(2, 2), bty = "n")



## 4.

### (a)

### The mean value of the recovery time provided by a complete case analysis is 19.27273.

### The associated standard error is 2.603013.

### The correlations between the recovery time and the dose is 0.2391256.

### The correlations between the recovery time and the blood pressure is -0.01952862.

load("databp.Rdata")  
recov<- which(is.na(databp$recovtime) == FALSE)  
merecov<-mean(databp$recovtime, na.rm = TRUE)  
serecov<-sd(databp$recovtime, na.rm = TRUE)/sqrt(length(recov))  
  
cor1<-cor(databp$recov, databp$logdose, use = "complete")  
cor2<-cor(databp$recov, databp$bloodp, use = "complete")  
  
merecov

## [1] 19.27273

serecov

## [1] 2.603013

cor1

## [1] 0.2391256

cor2

## [1] -0.01952862

### (b)

### The mean value of the recovery time provided by a complete case analysis is 19.27273.

### The associated standard error is 2.284135.

### The correlations between the recovery time and the dose is 0.2150612.

### The correlations between the recovery time and the blood pressure is -0.01934126.

re\_mi <- ifelse(is.na(databp$recovtime), mean(databp$recovtime, na.rm = TRUE), databp$recovtime)  
  
mearecov<-mean(re\_mi)  
searecov<-sd(re\_mi)/sqrt(length(re\_mi))  
cor1\_b<-cor(re\_mi, databp$logdose)  
cor2\_b<-cor(re\_mi, databp$bloodp)  
  
mearecov

## [1] 19.27273

searecov

## [1] 2.284135

cor1\_b

## [1] 0.2150612

cor2\_b

## [1] -0.01934126

### (c)

### The mean value of the recovery time provided by a complete case analysis is 19.44428.

### The associated standard error is 2.312845.

### The correlations between the recovery time and the dose is 0.2801835.

### The correlations between the recovery time and the blood pressure is -0.0111364.

fit\_c <- lm(recovtime ~ logdose+bloodp, data = databp)   
predicted\_ri <- predict(fit\_c, newdata = databp)  
re\_ri <- ifelse(is.na(databp$recovtime), predicted\_ri, databp$recovtime)  
  
merrecov<-mean(re\_ri)  
serrecov<-sd(re\_ri)/sqrt(length(re\_ri))  
cor1\_c<-cor(re\_ri, databp$logdose)  
cor2\_c<-cor(re\_ri, databp$bloodp)  
  
merrecov

## [1] 19.44428

serrecov

## [1] 2.312845

cor1\_c

## [1] 0.2801835

cor2\_c

## [1] -0.0111364

### (d)

### The mean value of the recovery time provided by a complete case analysis is 20.4598.

### The associated standard error is 2.444571.

### The correlations between the recovery time and the dose is 0.2284537.

### The correlations between the recovery time and the blood pressure is -0.01786944.

### As for stochastic regression imputation, with adding noise term, which can lead to relatively implausible predictions. For example, it makes no sense that impute a negative value for the Variables that are prone to missing.

fit\_d <- lm(recovtime ~ logdose+bloodp, data = databp)   
  
set.seed(1)  
predicted\_sri <- predict(fit\_d, newdata = databp) + rnorm(nrow(databp), 0, sigma(fit\_d))  
re\_sri <- ifelse(is.na(databp$recovtime), predicted\_sri, databp$recovtime)  
  
mesrecov<-mean(re\_sri)  
sesrecov<-sd(re\_sri)/sqrt(length(re\_sri))  
cor1\_d<-cor(re\_sri, databp$logdose)  
cor2\_d<-cor(re\_sri, databp$bloodp)  
  
mesrecov

## [1] 20.4598

sesrecov

## [1] 2.444571

cor1\_d

## [1] 0.2284537

cor2\_d

## [1] -0.01786944

### (e)

### As for this question, as shown in the output, first three lines output is the predicted values of each individuals in recovtime. Further, for the 4th, 10th and 22th missing value in recovtime should be imputed with the 6th, 2nd, and 17th value in the recovtime. And the imputed vales are 13, 10, 39 respectively. Finally, there is an output with the whole vales (after imputing) of recovtime.

f<-function(logdose,bloodp){  
 co<-coef(fit\_c)  
 print (co[[1]]+logdose\*co[[2]]+bloodp\*co[[3]])  
}  
  
pre\_value<-f(databp[,1],databp[,2])

## [1] 22.76836 21.50227 15.62087 14.26254 19.65178 15.44064 20.10469 22.55738  
## [9] 19.44950 21.51562 19.70013 20.64075 11.80965 20.15729 15.53733 19.87146  
## [17] 25.85865 21.42318 24.00371 22.60573 11.84059 26.32896 18.06486 19.28686  
## [25] 16.10433

y<-databp$recovtime  
  
for (i in 1:25){   
 if (((pre\_value[i]-pre\_value[4])^2) == min((pre\_value[-c(4,10,22)]-pre\_value[4])^2)){  
 print(i)  
 y[4]<-y[i]  
 print(y[4])  
 }  
}

## [1] 6  
## [1] 13

for (i in 1:25){   
 if (((pre\_value[i]-pre\_value[10])^2) == min((pre\_value[-c(4,10,22)]-pre\_value[10])^2)){  
 print(i)  
 y[10]<-y[i]  
 print(y[10])  
 }  
}

## [1] 2  
## [1] 10

for (i in 1:25){   
 if (((pre\_value[i]-pre\_value[22])^2) == min((pre\_value[-c(4,10,22)]-pre\_value[22])^2)){  
 print(i)  
 y[22]<-y[i]  
 print(y[22])  
 }  
}

## [1] 17  
## [1] 39

print(y)

## [1] 7 10 18 13 10 13 21 12 9 10 20 31 23 22 13 9 39 28 12 60 10 39 22 21 14

### (f)

#### Predictive mean matching over stochastic regression imputation will be more precision. And predictive mean matching can relatively fix the problem that stochastic regression attenuate statndard errors of the estimates as many single imputation.

### Firstly, there are many problems which are nonlinear, hence it will be more precision if this problem can be modified. Additionally, this approach maybe can produce substantially biased estimates of correlations and regression coefficients, which is another potential problem.