Incomplete Data Analysis

Assignment 1

Xuetao Liu s2257397

2022/2/2

Question 1

- (a) We should choose **(ii)0.3** as our answer. By assumption, ALQ is MCAR, that is to say the probability of ALQ is missing is completely at random. Therefore we have the following

$$P(\text{ALQ is missing}|\text{ALQ=Yes}) = P(\text{ALQ is missing}|\text{ALQ=No}) = P(\text{ALQ is missing}) = 0.3$$

- (b) We should choose **(ii)**. Since ALQ is MAR given gender means, so the probability of ALQ being missing is independent with the Yes/No value of the ALQ, but only depends on the gender.

- (c) We should choose **(iii)**. Since ALQ is MAR given gender, and we have the probability of ALQ being missing for men is 0.1, that is

$$P(\text{ALQ is missing}|\text{Men}) = 0.1.$$

  By definition, we have

$$P(\text{ALQ is missing}) = P(\text{ALQ is missing}|\text{Men}) + P(\text{ALQ is missing}|\text{Women}).$$

  But we do not know the probability of ALQ being missing, and thus we cannot know the probability of ALQ is missing given women.

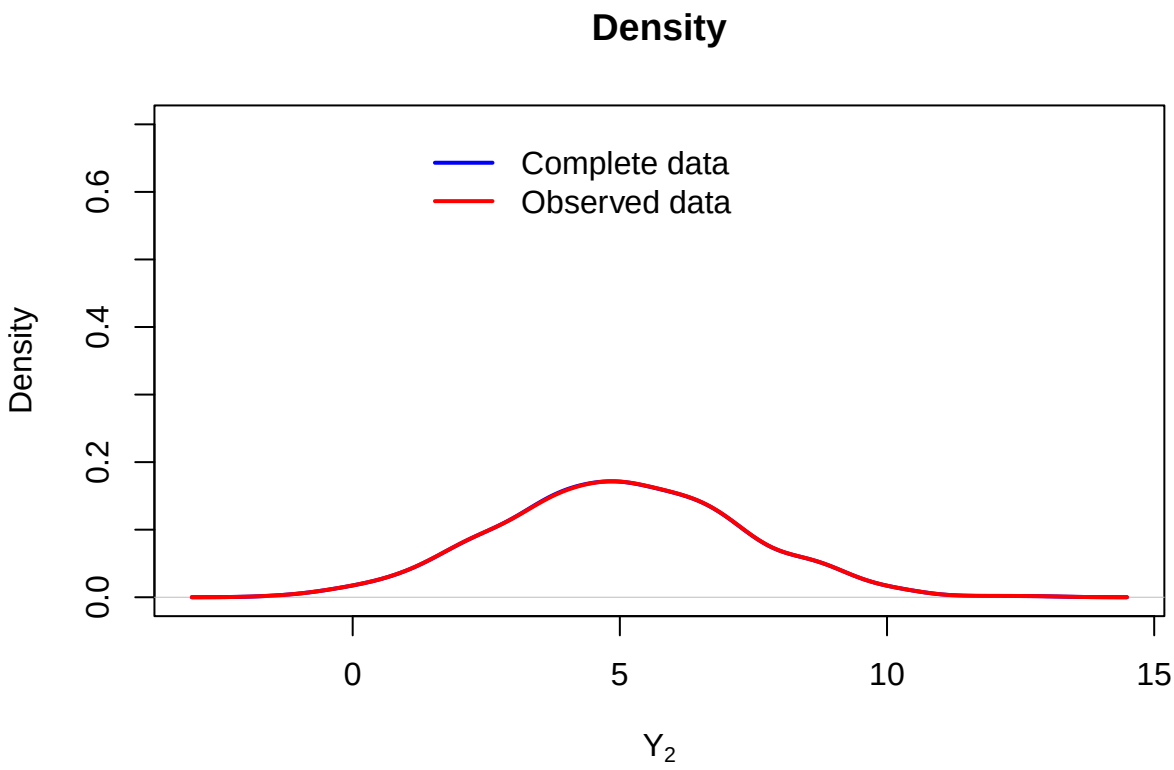
Question 2

- The largest possible subsample under a complete case analysis is that we have the 90% of the original data. In this case, the missing data are all in the same subjects, e.g. subjects 1-10 lost all the data for variable 1-10.
- The smallest subsample is that we have nothing left. In this case, the missing data has no intersection within subjects, e.g. subjects 1-10 lost variable 1, subjects 11-20 lost variable 2, ..., subjects 91-100 lost variable 10.

Question 3

(a)

```r
set.seed(1)
n = 500
mu = 0
sd = 1
a = 2
b = 0

## Generate the standard normal
z1 = rnorm(n,mu,sd)
z2 = rnorm(n,mu,sd)
z3 = rnorm(n,mu,sd)

y1 = 1 + z1
y2 = 5 + 2*z1 + z2

## According to the condition, a=2, b=0
## our condition becomes r = 2(y1-1), it is MAR
r = a*(y1-1) + b*(y2-5) + z3

# Find the observed and missing values as required
ind = r<0
y2_obs = y2[-ind]
y2_mis = y2[ind]

#plotting the densities
plot(density(y2), lwd = 2, col = "blue", xlab = expression(Y[2]),
main = "Density", ylim = c(0, 0.7))
lines(density(y2_obs), lwd = 2, col = "red")
#lines(density(y2_mis), lwd = 2, col = "darkgreen")
legend(1, 0.7, legend = c("Complete data", "Observed data"),
col = c("blue", "red"), lty = c(1,1), lwd = c(2,2), bty ="n")
```



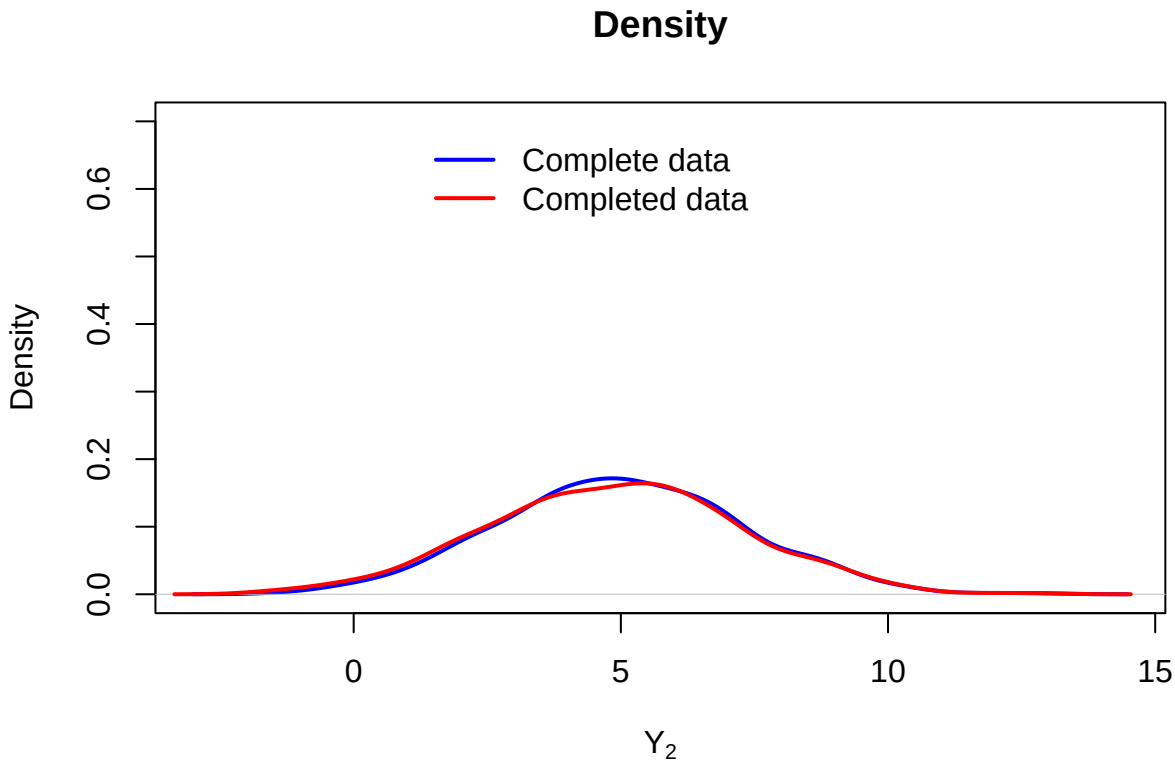
The above figure is the plot of density of complete data, observed data and missing data. By default, we set $a = 2$ and $b = 0$, which causes the missingness depends on the value of Y_1, which is **MAR**.

(b)

Here we use stochastic regression imputation, first we come to fit the regression model using the observed data

```
## ind is the index of those missing
## We can first create the data frame for regression
data = data.frame(y1, y2)
data$y2[ind] = NA # Impose the NA

fit = lm(y2 ~ y1, data = data)

# The predict values for the NAs
predicted_sri <- predict(fit, newdata = data) + rnorm(nrow(data), 0, sigma(fit))

# Impute the SRI
y2_sri <- ifelse(is.na(data$y2), predicted_sri, data$y2)

#plotting the densities
plot(density(y2), lwd = 2, col = "blue", xlab = expression(Y[2]),
main = "Density", ylim = c(0, 0.7))
lines(density(y2_sri), lwd = 2, col = "red")
```

```
legend(1, 0.7, legend = c("Complete data", "Completed data"),
col = c("blue", "red"), lty = c(1,1), lwd = c(2,2), bty ="n")
```

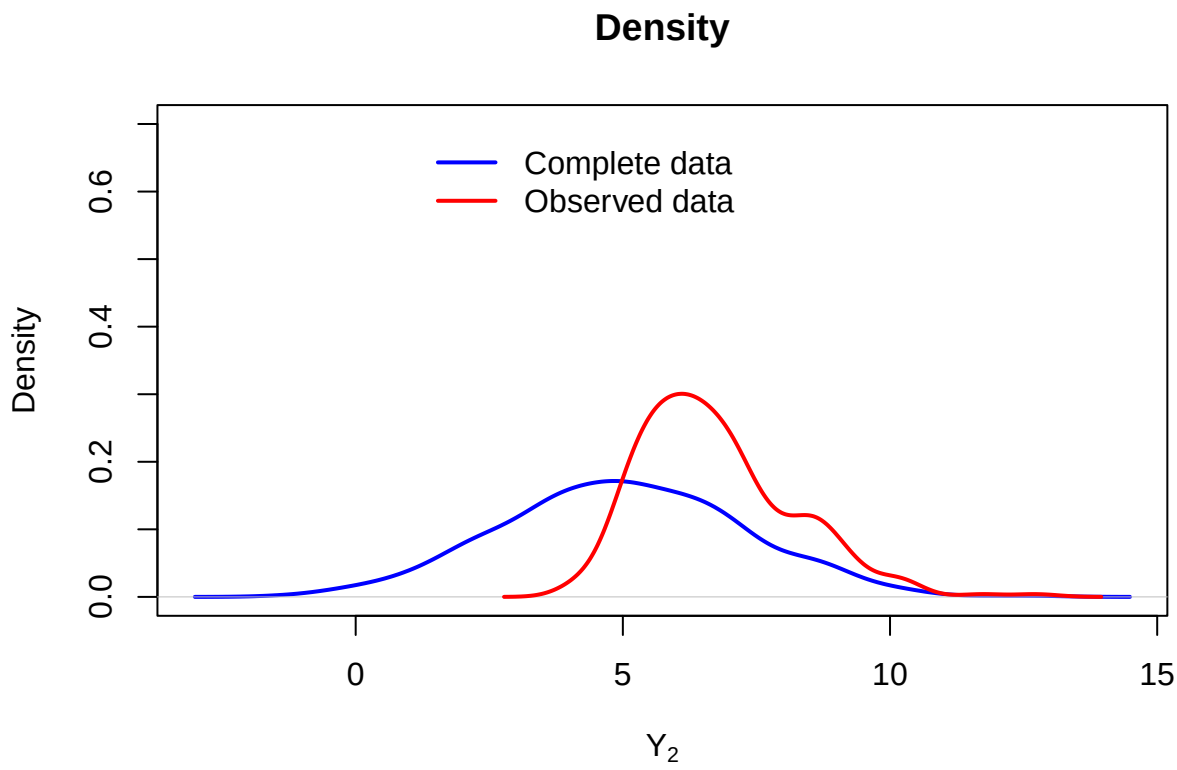


(c)

Here we take $a = 0$ and $b = 2$, which makes the missingness become **MNAR**

```
set.seed(1)
n = 500
mu = 0
sd = 1
a = 0
b = 2

## Generate the standard normal
z1 = rnorm(n,mu,sd)
z2 = rnorm(n,mu,sd)
z3 = rnorm(n,mu,sd)

y1 = 1 + z1
y2 = 5 + 2*z1 + z2

## According to the condition
r = a*(y1-1) + b*(y2-5) + z3
```

```r
# Find the indicator
ind = which(r<0)
y2_obs = y2[-ind]
y2_mis = y2[ind]

#plotting the densities
plot(density(y2), lwd = 2, col = "blue", xlab = expression(Y[2]),
main = "Density", ylim = c(0, 0.7))
lines(density(y2_obs), lwd = 2, col = "red")
#lines(density(y2_mis), lwd = 2, col = "darkgreen")
legend(1, 0.7, legend = c("Complete data", "Observed data"),
col = c("blue", "red"), lty = c(1,1,1), lwd = c(2,2,2), bty ="n")
```

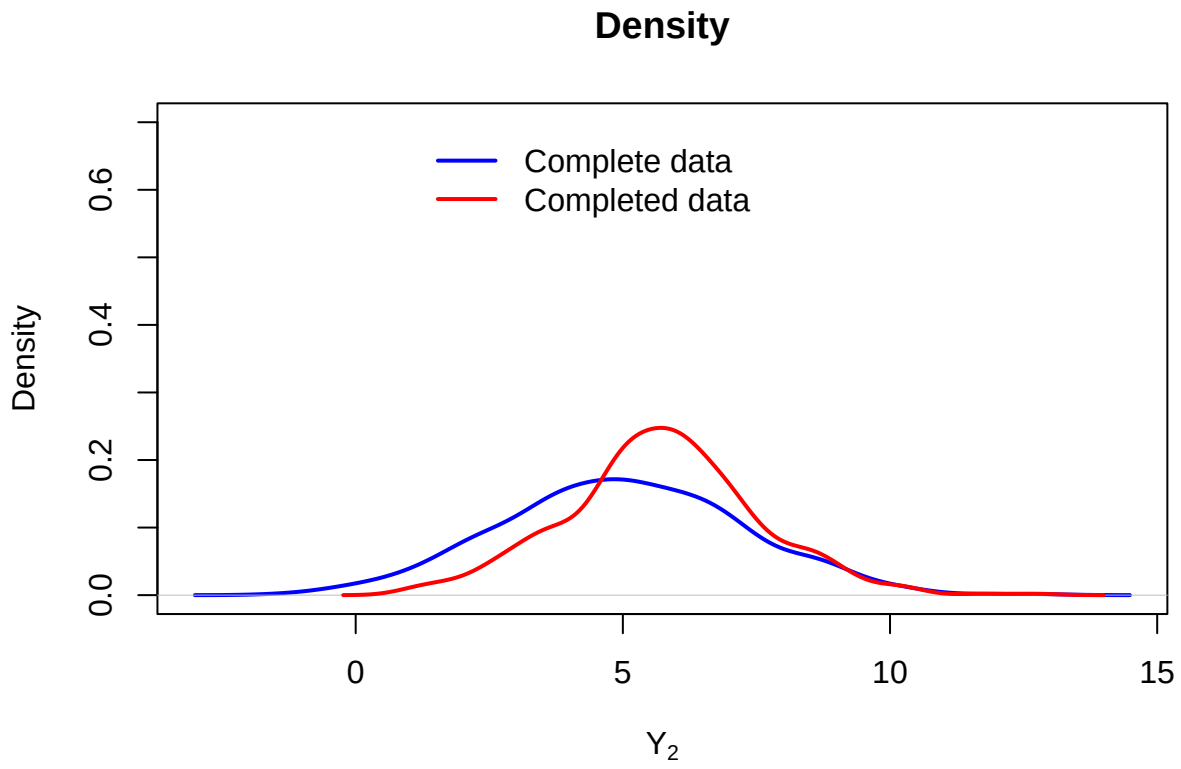


(d)

Like (b), we first create the dataframe, then fit the regression and then use the predicted value to conduct the stochastic regression imputation.

```r
## ind is the index of those missing
## We can first create the data frame for regression
data = data.frame(y1, y2)
data$y2[ind] = NA # Impose the NA

fit = lm(y2 ~ y1, data = data)
```

```r
# The predict values for the NAs
predicted_sri <- predict(fit, newdata = data) + rnorm(nrow(data), 0, sigma(fit))

# Impute the SRI
y2_sri <- ifelse(is.na(data$y2), predicted_sri, data$y2)

#plotting the densities
plot(density(y2), lwd = 2, col = "blue", xlab = expression(Y[2]),
main = "Density", ylim = c(0, 0.7))
lines(density(y2_sri), lwd = 2, col = "red")
legend(1, 0.7, legend = c("Complete data", "Completed data"),
col = c("blue", "red"), lty = c(1,1), lwd = c(2,2), bty ="n")
```



Question 4

(a)

First, we load the data

```r
load("databp.RData")
```

The mean and the associated standard error of the recovery time after removing the NAs is given as follow:

```r
# Number of non-missing values
n_obs = sum(is.na(databp$recovtime) == F)
# The mean of recovery time after removing the NAs
print(paste("The mean of original data is", mean(databp$recovtime, na.rm = T))) #19.27272727272273
## [1] "The mean of original data is 19.2727272727273"
print(paste("The standard error of original data is", sd(databp$recovtime, na.rm = T)/sqrt(n_obs))) #2.
## [1] "The standard error of original data is 2.60301342036263"
```

The Pearson correlations between the recovery time and the dose and between blood pressure is given as

```r
## Between recovery time and dose
cor(databp$recovtime, databp$logdose, use = "complete", method = "pearson") ##0.2391256
```

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

```r
## Between recovery time and bloop pressure
cor(databp$recovtime, databp$bloodp, use = "complete", method = "pearson") ##-0.01952862
```

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

(b)

To conduct mean imputation, we first need to calculate the mean value of recovery time

```r
m_recov = mean(databp$recovtime, na.rm = TRUE)
m_recov #19.27273
```

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

Then we substitute the NAs by the mean

```r
recov_mi <- ifelse(is.na(databp$recovtime), m_recov, databp$recovtime)
```

Since we just substitute the NAs by its mean, which would not change the global mean, so the mean after imputation remains the same

```r
n = length(recov_mi)
print(paste("The mean after mean imputation is", mean(recov_mi))) #19.27273
## [1] "The mean after mean imputation is 19.2727272727273"
print(paste("The standard error after mean imputation is", sd(recov_mi)/sqrt(n))) #2.284
## [1] "The standard error after mean imputation is 2.28413500635858"
```

We can see that the standard error after imputations is **2.284**, which is lower than the original **2.603**, which is the same as our conclusion that mean imputation would lower the standard error. The Pearson correlations between the recovery time after imputation and the dose is given by

```r
cor(recov_mi, databp$logdose, use = "complete", method = "pearson") ##0.2150612
```

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

We can see that the correlation also decreases from **0.2391256** to **0.2150612**. The Pearson correlations between the recovery time after imputation and the blood pressure is given as

```
cor(recov_mi, databp$bloodp, use = "complete", method = "pearson") ##-0.01952862
```

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

The correlation also decreases from **|-0.01952862|** to **|-0.01934126|**, since we only care about the magnitude.

(c)

To conduct the regression imputation, we first need to fit the regression model and then use the predicted value to substitute the NAs

```
fit = lm(recovtime ~ logdose + bloodp, data = databp)

# The predict values for the NAs
predicted_ri <- predict(fit, newdata = databp)

# Impute the SRI
recov_ri <- ifelse(is.na(databp$recovtime), predicted_ri, databp$recovtime)
```

Then for the mean and standard after regression imputation

```
print(paste("The mean after regression imputation is", mean(recov_ri))) #19.4442
## [1] "The mean after regression imputation is 19.4442847814827"
print(paste("The standard error after regression imputation is", sd(recov_ri)/sqrt(n))) #2.3128
## [1] "The standard error after regression imputation is 2.31284487743953"
```

The correlation after regression imputation is increased from **0.2150612** to **0.2801835**, which is also in the pattern of our conclusion of the regression imputation

```
cor(recov_ri, databp$logdose, use = "complete", method = "pearson") ## 0.2801
```

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

The Pearson correlations between the recovery time after imputation and the blood pressure is given as

```
cor(recov_ri, databp$bloodp, use = "complete", method = "pearson") ## - 0.0111
```

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

(d)

To conduct the stochastic regression imputation, we can use the model we build above, and add the residual terms to add the variability.

```r
# The predict values for the NAs
predicted_sri <- predict(fit, newdata = databp) + rnorm(nrow(databp), 0, sigma(fit))

# Impute the SRI
recov_sri <- ifelse(is.na(databp$recovtime), predicted_sri, databp$recovtime)
```

Then for the mean and standard after stochastic regression imputation

```r
print(paste("The mean after stochastic regression imputation is", mean(recov_sri))) # 18.9825
## [1] "The mean after stochastic regression imputation is 19.1863842537346"
print(paste("The standard error after stochastic regression imputation is", sd(recov_sri)/sqrt(n))) # 2
## [1] "The standard error after stochastic regression imputation is 2.28850997711992"
```

The correlation after stochastic regression imputation has dropped from **0.2150612** to **0.05836918**

```r
cor(recov_sri, databp$logdose, use = "complete", method = "pearson") # 0.05836918
```

```
## [1] 0.189576
```

The Pearson correlations between the recovery time after imputation and the blood pressure is given as

```r
cor(recov_sri, databp$bloodp, use = "complete", method = "pearson") ## -0.04563002
```

```
## [1] -0.02117405
```

(e)

Here we conduct the mean matching method to substitute the missing values. First we can use the predicted value in part (c)

```r
recov_ri
```

```
##  [1]  7.00000 10.00000 18.00000 14.26254 10.00000 13.00000 21.00000 12.00000
##  [9]  9.00000 21.51562 20.00000 31.00000 23.00000 22.00000 13.00000  9.00000
## [17] 39.00000 28.00000 12.00000 60.00000 10.00000 26.32896 22.00000 21.00000
## [25] 14.00000
```

```r
# Obtain the index of NAs
ind = which(is.na(databp$recovtime) == T)
ind ## The index of the missing values
```

```
## [1]  4 10 22
```

```r
recov_ri[ind]
```

```
## [1] 14.26254 21.51562 26.32896
```

```
recov_obs_pre = predicted_ri[-ind] ## predicted value for the observed subjects
recov_obs = recov_ri[-ind] ## Observed values
recov_pre = recov_ri[ind] ## Predicted values for the missing subjects
```

What we need to do then is to find the closest number in the observed value (measured in squared difference) as the donator for the missing values

```
## Number of missing values
n = length(ind)
## Initialize the vector to store the index of the donars
idoner = numeric(n)
for (i in 1:n){
  idoner[i] = which.min((recov_obs_pre-recov_pre[i])^2)
}

doner = recov_obs[idoner]
```

And we can compare the donars to our original predicted values to see whether they are close

```
doner # 13 10 39
```

```
## [1] 13 10 39
```

```
recov_pre # 14.26254 21.51562 26.32896
```

```
## [1] 14.26254 21.51562 26.32896
```

So we replace the predicted value by the donars

```
recov_pmm = recov_ri
recov_pmm[ind] = doner
recov_pmm
```

```
##  [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
```

Then for the mean and standard after predicted mean matching

```
print(paste("The mean after stochastic regression imputation is", mean(recov_pmm))) # 19.44
## [1] "The mean after stochastic regression imputation is 19.44"
print(paste("The standard error after stochastic regression imputation is", sd(recov_pmm)/sqrt(n))) # 2
## [1] "The standard error after stochastic regression imputation is 7.11430483837552"
```

The correlation after predicted mean matching is

```
cor(recov_pmm, databp$logdose, use = "complete", method = "pearson") # 0.3038
```

```
## [1] 0.3037945
```

The Pearson correlations between the recovery time after imputation and the blood pressure is given as

```r
cor(recov_pmm, databp$bloodp, use = "complete", method = "pearson") ## -0.0321
```

```
## [1] -0.03208685
```

(f)

For the predictive mean matching method, all the missing values are substituted by the observed values, which can avoid the implausible values.

One of the potential problem is that when there are too many missing values(sparse) or the size of our data set is too small, the standard error of the results would be extremely small, which lost the variability of the data.