A3

Zihao Zhang

```
library(tidyverse)
library(data.table)
getwd()
setwd("/Users/zhangzihao/Desktop/duke/613/HW/a3/Data")
datstu <- fread("datstu_v2.csv")
datsss <- fread("datsss.csv")
datjss <- fread("datjss.csv")
```

```
# 1
# 1.1
programs <- datstu[,11:16] %>% unlist() %>% unique()
length(programs) #33
nrow(datstu) #340823
unique(datsss$schoolcode) %>% length() #898
```

```
> length(programs) #33
[1] 33
> nrow(datstu) #340823
[1] 340823
> unique(datsss$schoolcode) %>% length() #898
[1] 898
```

```
# 1.2
c1 <- datstu %>% select(schoolcode1, choicepgm1)
c2 <- datstu %>% select(schoolcode2, choicepgm2)
c3 <- datstu %>% select(schoolcode3, choicepgm3)
c4 <- datstu %>% select(schoolcode4, choicepgm4)
c5 <- datstu %>% select(schoolcode5, choicepgm5)
c6 <- datstu %>% select(schoolcode6, choicepgm6)
rbind(c1,c2,c3,c4,c5,c6,use.names=F) %>% unique() %>% nrow() #3086
```

```
> rbind(c1,c2,c3,c4,c5,c6,use.names=F) %>% unique() %>% nrow() #3086
[1] 3086
```

```
# 1.3

sch_dis_1 <- datsss[,3:4]
sch_dis_1 <- sch_dis_1[!duplicated(sch_dis_1$schoolcode),]
sch_dis_2 <- datstu[,c(5:10, 17)]
sch_dis_2 <- pivot_longer(sch_dis_2, !jssdistrict, values_to = "schoolcode")
sch_dis <- left_join(sch_dis_2,sch_dis_1,by="schoolcode")

sch_dis$same_dis = 0
for (i in 1:nrow(sch_dis)){
  if (isTRUE(c(sch_dis$jssdistrict[i]) == c(sch_dis$sssdistrict[i]))){
    sch_dis$same_dis[i] = 1
    }
}

sch_dis$number_same = 0
for (i in 1:340823){
  sch_dis$number_same[c((i-1)*6+1)] = sum(sch_dis$same_dis[c((i-1)*6+1):c(i*6)])
}
```

```
a <- filter(sch_dis,sch_dis$number_same >= 2)
nrow(a)
```

```
> nrow(a)
[1] 199208
```

```r
# 1.4
admit <- datstu
admit <- cbind(admit,datstu[,1])
admit <- admit[order(rankplace,)]
x <- matrix(0,nrow = nrow(admit))
admit <- cbind(admit,x)
admit <- admit[,-19]
colnames(admit)[19] <- c("school_admit")
admit <- admit %>% na.omit()
count <- admit %>% group_by(rankplace) %>% count()
for (i in 1:42361){
  admit[i,19] = admit[i,1]
}
for (i in 42362:72822){
  admit[i,19] = admit[i,2]
}
for (i in 72823:100658){
  admit[i,19] = admit[i,3]
}
for (i in 100659:124926){
  admit[i,19] = admit[i,4]
```

```r
 for (i in 124927:128801){
   admit[i,19] = admit[i,5]
 }
 for (i in 128802:131670){
   admit[i,19] = admit[i,6]
 }
 number_adimit <- admit %>% group_by(school_admit) %>% count()
 number_adimit
```

school_admit	n
0	20227
10101	374
10102	220
10103	389
10104	209
10105	324
10106	359
10107	288
10108	292
10109	283
10110	447
10111	520
10112	274
10114	318

```r
#1.5
admit_sch <- admit[,8:9]
cut_off <- left_join(datstu,admit_sch,by = "V1") %>% na.omit()
cut_off <- cut_off[,c(2,19)]
cut_off %>% group_by(school_admit) %>% summarise(min(score))

# 1.6
cut_off %>% group_by(school_admit) %>% summarise(mean(score))
```

```
> cut_off %>% group_by(school_admit) %>% summarise(min(score))
# A tibble: 518 x 2
   school_admit `min(score)`
          <dbl>        <int>
 1            0          192
 2        10101          284
 3        10102          343
 4        10103          316
 5        10104          245
 6        10105          260
 7        10106          293
 8        10107          281
 9        10108          248
10        10109          257
# ... with 508 more rows
```

```
# A tibble: 518 × 2
   school_admit `mean(score)`
          <dbl>         <dbl>
 1            0          259.
 2        10101          320.
 3        10102          394.
 4        10103          354.
 5        10104          297.
 6        10105          351.
 7        10106          340.
 8        10107          312.
 9        10108          303.
10        10109          282.
# … with 508 more rows
```

```r
# 2
# we need to calculate how many students each program admitted
# I will use the old method to construct a more specific list
pro <- datstu[,c(2,5:16,18)]
pro <- cbind(pro,datstu[,1])
pro <- pro[order(rankplace,)]
x <- matrix(0,nrow = nrow(pro))
pro <- cbind(pro,x)
colnames(pro)[16] <- c("school_admit")
pro <- pro %>% na.omit()
count <- pro %>% group_by(rankplace) %>% count()
for (i in 1:as.numeric(count[1,2])){
  pro[i,16] = pro[i,2]
}
for (i in as.numeric(count[1,2]+1):as.numeric(count[1,2]+count[2,2])){
  pro[i,16] = pro[i,3]
}
for (i in as.numeric(count[2,2]+1):as.numeric(count[1,2]+count[2,2]+count[3,2])){
  pro[i,16] = pro[i,4]
}
```

```r
for (i in as.numeric(count[3,2]+1):as.numeric(count[1,2]+count[2,2]+count[3,2]+count[4,2])){
  pro[i,16] = pro[i,5]
}
for (i in as.numeric(count[4,2]+1):as.numeric(count[1,2]+count[2,2]+count[3,2]+count[4,2]+count[5,2])){
  pro[i,16] = pro[i,6]
}
for (i in as.numeric(count[5,2]+1):as.numeric(count[1,2]+count[2,2]+count[3,2]+count[4,2]+count[5,2]+count[6,2])){
  pro[i,16] = pro[i,7]
}

# I slightly improve the last method, then is "program"
y <- matrix("NA",nrow = nrow(pro))
pro <- cbind(pro,y)
colnames(pro)[17] <- c("program_admit")
for (i in 1:as.numeric(count[1,2])){
  pro[i,17] = pro[i,8]
}
for (i in as.numeric(count[1,2]+1):as.numeric(count[1,2]+count[2,2])){
  pro[i,17] = pro[i,9]
}
```

```r
for (i in as.numeric(count[2,2]+1):as.numeric(count[1,2]+count[2,2]+count[3,2])){
  pro[i,17] = pro[i,10]
}
for (i in as.numeric(count[3,2]+1):as.numeric(count[1,2]+count[2,2]+count[3,2]+count[4,2])){
  pro[i,17] = pro[i,11]
}
for (i in as.numeric(count[4,2]+1):as.numeric(count[1,2]+count[2,2]+count[3,2]+count[4,2]+count[5,2])){
  pro[i,17] = pro[i,12]
}
for (i in as.numeric(count[5,2]+1):as.numeric(count[1,2]+count[2,2]+count[3,2]+count[4,2]+count[5,2]+count[6,2])){
  pro[i,17] = pro[i,13]
}

new_pro <- pro[,c(1,15:17)]
new_pro <- new_pro %>% na.omit()
new_pro <- new_pro[!apply(new_pro == "", 1, all),] # delete the blank
new_pro <- new_pro %>% mutate(sch_pro = paste0(school_admit,sep = ",",program_admit))
colnames(new_pro)[3] <- "schoolcode"
new_pro2 <- left_join(new_pro,datsss,by = "schoolcode") %>% na.omit()
new_pro2 <- new_pro2[!apply(new_pro2 == "", 1, all),] # delete the blank
```

```r
new_an <- new_pro2 %>% group_by(sch_pro) %>% count()
colnames(new_an) <- c("sch_pro","admit_number")
new_pro2 <- left_join(new_pro2,new_an,by = "sch_pro")

cutoff <- new_pro %>% group_by(sch_pro) %>% summarise(min(score))
cutoff <- cutoff[-1,]
new_pro2 <- left_join(new_pro2,cutoff,by = "sch_pro")

quality <- new_pro %>% group_by(sch_pro) %>% summarise(mean(score))
quality <- quality[-1,]
new_pro2 <- left_join(new_pro2,quality,by = "sch_pro")

school <- new_pro2[,c(5,7:13)]
school <- school %>% group_by(sch_pro) %>% filter (!duplicated(sch_pro))
```

lname	sssdistrict	ssslong	ssslat	admit_number	min(score)	mean(score)
:AMPA SENIOR HIGH TECHNICAL	Abura/Asebu/Kwamankese (Abura Dunkwa)	-1.19708836	5.130001	160	208	251.9250
:AMPA SENIOR HIGH TECHNICAL	Abura/Asebu/Kwamankese (Abura Dunkwa)	-1.19708836	5.130001	32	224	245.2500
:AMPA SENIOR HIGH TECHNICAL	Abura/Asebu/Kwamankese (Abura Dunkwa)	-1.19708836	5.130001	104	216	254.7692
:I PRESBY SENIOR HIGH. SCHOOL, ABETIFI	Kwahu South (Mpraeso)	-0.63552868	6.619226	345	269	304.1449
:I PRESBY SENIOR HIGH. SCHOOL, ABETIFI	Kwahu South (Mpraeso)	-0.63552868	6.619226	140	272	301.1071
:I PRESBY SENIOR HIGH. SCHOOL, ABETIFI	Kwahu South (Mpraeso)	-0.63552868	6.619226	115	257	297.4348
:I PRESBY SENIOR HIGH. SCHOOL, ABETIFI	Kwahu South (Mpraeso)	-0.63552868	6.619226	110	252	288.1818
:I PRESBY SENIOR HIGH. SCHOOL, ABETIFI	Kwahu South (Mpraeso)	-0.63552868	6.619226	90	254	290.0556
:I TECH. INST., ABETIFI	Kwahu South (Mpraeso)	-0.63552868	6.619226	140	204	238.4000
:I TECH. INST., ABETIFI	Kwahu South (Mpraeso)	-0.63552868	6.619226	14	246	246.0000
SENIOR HIGH SCHOOL, ABOR	Keta	0.85305578	5.907464	248	216	256.6290
SENIOR HIGH SCHOOL, ABOR	Keta	0.85305578	5.907464	592	207	269.0068
SENIOR HIGH SCHOOL, ABOR	Keta	0.85305578	5.907464	708	211	267.9548
SENIOR HIGH SCHOOL, ABOR	Keta	0.85305578	5.907464	240	214	264.6000

```r
# 3
sch <- admit
colnames(sch)[19] <- "schoolcode"
distant <- left_join(sch,datsss,by="schoolcode")
distant <- select(distant,c("V1.x","schoolcode","jssdistrict"))
distant <- distant[!duplicated(distant$schoolcode),]
colnames(distant)[1] <- "student"
y <- datjss
distant <- left_join(distant,y,by = "jssdistrict")
colnames(distant)[5] <- "jsslong"
colnames(distant)[6] <- "jsslat"
distant <- left_join(distant,datsss,by = "schoolcode")
distant <- distant %>% select(ssslong,jsslong,jsslat,ssslat)
distant <- distant %>%
  mutate(dist = sqrt( (69.172*(ssslong - jsslong) * cos(jsslat/57.3)) ^2  +  (69.172 * (ssslat - jsslat))^2 ))
distant <- na.omit(distant)
distant
```

	ssslong	jsslong	jsslat	ssslat	dist
1	-1.1970884	-1.00538456	5.401725	5.130001	22.96873
2	-1.1970884	-1.00538456	5.401725	5.130001	22.96873
3	-1.1970884	-1.00538456	5.401725	5.130001	22.96873
4	-1.1970884	-1.00538456	5.401725	5.130001	22.96873
5	-0.6355287	-0.47498974	5.944515	6.619226	47.96028
6	-0.6355287	-0.47498974	5.944515	6.619226	47.96028
7	-0.6355287	-0.47498974	5.944515	6.619226	47.96028
8	-0.6355287	-0.47498974	5.944515	6.619226	47.96028
9	-0.6355287	-0.47498974	5.944515	6.619226	47.96028
10	-0.6355287	-0.35609409	6.436071	6.619226	23.00930
11	-0.6355287	-0.35609409	6.436071	6.619226	23.00930
12	-0.6355287	-0.35609409	6.436071	6.619226	23.00930
13	-0.6355287	-0.35609409	6.436071	6.619226	23.00930
14	-0.6355287	-0.35609409	6.436071	6.619226	23.00930

```r
# 4
# 4.1
scode_rev <- datsss[,3]
scode_rev <- scode_rev[!duplicated(scode_rev$schoolcode),]
scode_rev$scode_rev <- substr(scode_rev$schoolcode, 1, 3)

# it is a list. then we may change the data in the datstu

datstu_4 <- datstu
datstu_4$scode_rev1 <- substr(datstu_4$schoolcode1, 1, 3)
datstu_4$scode_rev2 <- substr(datstu_4$schoolcode2, 1, 3)
datstu_4$scode_rev3 <- substr(datstu_4$schoolcode3, 1, 3)
datstu_4$scode_rev4 <- substr(datstu_4$schoolcode4, 1, 3)
datstu_4$scode_rev5 <- substr(datstu_4$schoolcode5, 1, 3)
datstu_4$scode_rev6 <- substr(datstu_4$schoolcode6, 1, 3)
```

scode_rev1	scode_rev2	scode_rev3	scode_rev4	scode_rev5	scode_rev6	pgm_rev6	pgm_rev5	pgm_re
501	501	502	502	507	509	arts	economics	arts
701	706	701	701	706	706	arts	economics	arts
507	507	501	507	516	507	economics	economics	econor
905	904	901	909	901	903	arts	others	others
518	517	502	502	516	502	economics	arts	arts
101	501	517	502	506	516	economics	economics	arts
803	804	803	804	805	809	arts	arts	arts
403	404	404	403	402	403	others	others	others
213	213	212	212	202	201	arts	arts	science
801	904	505	509	505	505	arts	arts	arts
518	506	505	506	506	509	arts	arts	econor
100	905	801	905	902	906	others	science	science
306	306	309	309	306	309	arts	economics	econor
801	801	801	801	804	810	others	others	econor

```r
# 4.2
science <- c("General Science")
arts <- c("General Arts", "Visual Arts")
economics <- c("Business", "Home Economics")

datstu_4 <- within(datstu_4, {
  pgm_rev1 = "others"
  pgm_rev1[choicepgm1 %in% arts] = "arts"
  pgm_rev1[choicepgm1 %in% economics] = "economics"
  pgm_rev1[choicepgm1 %in% science] = "science"
  pgm_rev1[is.na(pgm_rev1) == T] = "others"

  pgm_rev2 = "others"
  pgm_rev2[choicepgm2 %in% arts] = "arts"
  pgm_rev2[choicepgm2 %in% economics] = "economics"
  pgm_rev2[choicepgm2 %in% science] = "science"
  pgm_rev2[is.na(pgm_rev2) == T] = "others"
```

```r
  pgm_rev3 = "others"
  pgm_rev3[choicepgm3 %in% arts] = "arts"
  pgm_rev3[choicepgm3 %in% economics] = "economics"
  pgm_rev3[choicepgm3 %in% science] = "science"
  pgm_rev3[is.na(pgm_rev3) == T] = "others"

  pgm_rev4 = "others"
  pgm_rev4[choicepgm4 %in% arts] = "arts"
  pgm_rev4[choicepgm4 %in% economics] = "economics"
  pgm_rev4[choicepgm4 %in% science] = "science"
  pgm_rev4[is.na(pgm_rev4) == T] = "others"

  pgm_rev5 = "others"
  pgm_rev5[choicepgm5 %in% arts] = "arts"
  pgm_rev5[choicepgm5 %in% economics] = "economics"
  pgm_rev5[choicepgm5 %in% science] = "science"
  pgm_rev5[is.na(pgm_rev5) == T] = "others"
```

```
  pgm_rev6 = "others"
  pgm_rev6[choicepgm6 %in% arts] = "arts"
  pgm_rev6[choicepgm6 %in% economics] = "economics"
  pgm_rev6[choicepgm6 %in% science] = "science"
  pgm_rev6[is.na(pgm_rev6) == T] = "others"
})
```

pgm_rev5	pgm_rev4	pgm_rev3	pgm_rev2	pgm_rev1	choice_rev1	choice_rev2	choice_rev3	choice_rev4
economics	arts	arts	arts	economics	501,economics	501,arts	502,arts	502,arts
economics	arts	arts	economics	arts	701,arts	706,economics	701,arts	701,arts
economics	economics	economics	economics	economics	507,economics	507,economics	501,economics	507,econom
others	others	others	arts	arts	905,arts	904,arts	901,others	909,others
arts	arts	economics	arts	economics	518,economics	517,arts	502,economics	502,arts
economics	arts	arts	arts	arts	101,arts	501,arts	517,arts	502,arts
arts	arts	arts	arts	arts	803,arts	804,arts	803,arts	804,arts
others	others	arts	arts	arts	403,arts	404,arts	404,arts	403,others
arts	science	science	economics	economics	213,economics	213,economics	212,science	212,science
arts	arts	arts	arts	arts	801,arts	904,arts	505,arts	509,arts
arts	economics	economics	arts	economics	518,economics	506,arts	505,economics	506,econom
science	science	science	science	science	100,science	905,science	801,science	905,science
economics	economics	arts	others	economics	306,economics	306,others	309,arts	309,econom
others	economics	economics	economics	economics	801,economics	801,economics	801,economics	801,econom

```
# 4.3
datstu_4 <- datstu_4 %>% mutate(choice_rev1 = paste0(scode_rev1,sep = ",",pgm_rev1)) %>%
  mutate(choice_rev2 = paste0(scode_rev2,sep = ",",pgm_rev2)) %>%
  mutate(choice_rev3 = paste0(scode_rev3,sep = ",",pgm_rev3)) %>%
  mutate(choice_rev4 = paste0(scode_rev4,sep = ",",pgm_rev4)) %>%
  mutate(choice_rev5 = paste0(scode_rev5,sep = ",",pgm_rev5)) %>%
  mutate(choice_rev6 = paste0(scode_rev6,sep = ",",pgm_rev6))
```

choice_rev1	choice_rev2	choice_rev3	choice_rev4	choice_rev5	choice_rev6
501,economics	501,arts	502,arts	502,arts	507,economics	509,arts
701,arts	706,economics	701,arts	701,arts	706,economics	706,arts
507,economics	507,economics	501,economics	507,economics	516,economics	507,economics
905,arts	904,arts	901,others	909,others	901,others	903,arts
518,economics	517,arts	502,economics	502,arts	516,arts	502,economics
101,arts	501,arts	517,arts	502,arts	506,economics	516,economics
803,arts	804,arts	803,arts	804,arts	805,arts	809,arts
403,arts	404,arts	404,arts	403,others	402,others	403,others
213,economics	213,economics	212,science	212,science	202,arts	201,arts
801,arts	904,arts	505,arts	509,arts	505,arts	505,arts
518,economics	506,arts	505,economics	506,economics	506,arts	509,arts
100,science	905,science	801,science	905,science	902,science	906,others
306,economics	306,others	309,arts	309,economics	306,economics	309,arts
801,economics	801,economics	801,economics	801,economics	804,others	810,others

```r
# 4.4
datstu4 <- datstu_4 %>% select(V1,score,rankplace,choice_rev1,choice_rev2,choice_rev3,choice_rev4,choice_rev5,choice_rev6)
datstu4 <- datstu4[order(rankplace,)]
count <- datstu4 %>% group_by(rankplace) %>% count()
for (i in 1:as.numeric(count[1,2])){
  datstu4$admit[i] = datstu4$choice_rev1[i]
}
for (i in as.numeric(count[1,2]+1):as.numeric(count[1,2]+count[2,2])){
  datstu4$admit[i] = datstu4$choice_rev2[i]
}
for (i in as.numeric(count[2,2]+1):as.numeric(count[1,2]+count[2,2]+count[3,2])){
  datstu4$admit[i] = datstu4$choice_rev3[i]
}
for (i in as.numeric(count[3,2]+1):as.numeric(count[1,2]+count[2,2]+count[3,2]+count[4,2])){
  datstu4$admit[i] = datstu4$choice_rev4[i]
}
for (i in as.numeric(count[4,2]+1):as.numeric(count[1,2]+count[2,2]+count[3,2]+count[4,2]+count[5,2])){
  datstu4$admit[i] = datstu4$choice_rev5[i]
```

```r
for (i in as.numeric(count[5,2]+1):as.numeric(count[1,2]+count[2,2]+count[3,2]+count[4,2]+count[5,2]+count[6,2])){
  datstu4$admit[i] = datstu4$choice_rev6[i]
}

cutoff <-datstu4 %>% group_by(admit) %>% summarise(min(score))
cutoff <- cutoff[-1,]
datstu_new <- left_join(datstu4,cutoff,by = "admit")

quality <-datstu_new %>% group_by(admit) %>% summarise(mean(score))
quality <- quality[-1,]
datstu_new <- left_join(datstu_new,quality,by = "admit")
datstu_new <- datstu_new %>% group_by(admit) %>% filter (!duplicated(admit))
new_cq <- datstu_new[,10:12] %>% na.omit()
```

	admit	min(score)	mean(score)
1	304,economics	192	284.5549
2	304,others	216	290.1190
3	210,arts	198	314.3445
4	210,economics	201	313.0429
5	210,science	218	342.4231
6	902,others	204	253.2619
7	705,economics	196	273.8411
8	705,arts	190	274.8638
9	213,arts	204	296.1403
10	213,economics	206	294.9786
11	213,others	206	283.9068
12	213,science	216	327.0606
13	203,arts	208	322.6435
14	203,economics	199	322.2130

```r
# 4.5
datstu5 <- datstu_4[order(-score), ]
datstu5 <- datstu5[1:20000,]
```

	V1	score	agey	male	schoolcode1	schoolcode2	schoolcode3	schoolcode4	schoolcode5	schoolcode6	choicepgm1
	335624	469	15	0	30107	30107	50102	21501	10403	10119	General Science
	318458	468	15	1	21003	40107	30106	10201	*NA*	*NA*	General Science
	318492	467	15	1	21003	20102	21302	20402	10504	21503	General Science
	335584	467	15	0	30107	21103	20301	21501	*NA*	*NA*	General Science
	318422	466	15	1	21003	20104	21303	20402	*NA*	*NA*	General Science
	318525	466	15	1	21003	20102	10105	21302	20603	21006	General Science
	335568	465	14	0	30107	50201	50102	10112	*NA*	*NA*	General Science
	335629	465	15	0	30107	20301	30301	30106	21007	20603	General Science
	335722	465	15	0	30107	30107	20301	20301	10205	10210	General Science
	239799	464	14	0	30103	10102	10202	10117	10205	10116	General Science
	268535	464	15	1	30104	30102	30905	30601	10210	10203	General Science
	289149	464	15	1	50110	20102	40104	30401	20105	21006	General Science
	335866	464	16	0	30107	30107	21103	21103	*NA*	*NA*	General Science
	335901	464	15	0	30107	30107	30107	30107	*NA*	*NA*	General Science

```r
# 5
data <- datstu5
data$choice_rev1 <- as.numeric(as.factor(data$choice_rev1))
score <- select(data,score)
choice <- select(data,choice_rev1)
n_choice <- choice[!duplicated(choice$choice_rev1),]
nrow(n_choice)
score <- as.matrix(score)

like_fun <- function(par,score,choice,data){
  n_i = nrow(score)
  n_j = nrow(n_choice)
  out = mat.or.vec(n_i,n_j)

  par1 = par[1:n_j-1]
  par2 = par[n_j:(2*n_j-2)]

  out[,1] = 0
```

```r
  for (j in 1:(n_j-1)){
    out[,j+1] = par1[j] + par2[j] * score
  }

  prob = exp(out)
  prob = sweep(prob, MARGIN = 1, STATS = rowSums(prob), FUN = "/")

  prob_choice = NULL

  for (z in 1:n_i){
    prob_choice[z] = prob[z,]
  }
  prob_choice[prob_choice>0.99999] = 0.99999
  prob_choice[prob_choice<0.00001] = 0.00001

  like = sum(log(prob_choice))
  return(-like)
}
```

```r
# Estimate parameters and compute the marginal effect of the proposed model
# Estimate parameters
n <- runif(490,-1,1)
result = optim(n,
               fn=like_fun,
               method = 'BFGS',
               control = list(trace = 6, maxit = 3000),
               score = score,
               choice = choice,
               data = data
               )

out_logit <- result
result$par
result$value
```

```
result$par
[1]  0.4945196491  0.5605091848  0.2928970144  0.5806552824 -0.0555830747  0.6511623994  0.9730458325 -0.3896163432
[9] -0.5300413053 -0.3564809901 -0.7513058279  0.1442976347 -0.8551723124  0.7931857882  0.4801902627  0.7060436746
17]  0.5892855735  0.8866877416 -0.8337877905 -0.0648775720  0.7181951189  0.4782690625 -0.7915115487  0.6625639135
25]  0.1571571408 -0.6124679549 -0.0281252395  0.7872906500 -0.2128377226 -0.3483333485  0.5437121391  0.9544018195
33] -0.2152902232 -0.3816896644  0.7982811518 -0.5943525420 -0.5138788382  0.1952546821  0.8518465455 -0.6884270818
41] -0.0487754522 -0.0189024732 -0.3724321234  0.0891752383  0.4343819218 -0.3441586699  0.0251909331 -0.8098731111
49]  0.4254824668  0.5845241062  0.8572961828 -0.7421781723  0.9907588339 -0.7089373106  0.2728602001  0.9374230225
57] -0.6911598104  0.5171433594 -0.0242504324 -0.2004548637  0.2446475821  0.5873536882  0.0062009492  0.7267738897
65]  0.7721571801 -0.7244460830  0.0253010886 -0.3955216985  0.2443955899  0.1531632482 -0.9970123991  0.6959534329
73]  0.6590137552 -0.9949057982  0.1196605130  0.3418226675 -0.2784064934 -0.6331207096 -0.9820895838 -0.6495441333
81]  0.8763925033 -0.7167485487  0.1617359254 -0.8260560273  0.6494959919  0.5784652592  0.2997369231  0.2925397628
89]  0.9569419860 -0.0963890930 -0.8257877775  0.2946580518  0.2246622415  0.0807685386 -0.8131091250 -0.9190669064
```

```
> result$value
[1] 230258.5
```

```r
# Marginal effect
prob_fun <- function(par,score,choice,data){
  n_i = nrow(score)
  n_j = nrow(n_choice)
  out = mat.or.vec(n_i,n_j)

  par1 = par[1:n_j-1]
  par2 = par[n_j:(2*n_j-2)]

  out[,1] = 0

  for (j in 2:(n_j-1)){
    out[,j+1] = par1[j] + par2[j] * score
  }

  return(out)
  }
```

```r
param <- result$par
prob <- prob_fun(param, score, choice, data)
prob <-  exp(prob)
prob <-  sweep(prob, MARGIN = 1, STATS = rowSums(prob), FUN = "/")
b_i_bar <- apply(prob,1,function(x) return(sum(b_j*x)))
Mar_e <- data.frame(prob *(b_j - b_i_bar) )
apply(Mar_e, MARGIN = 2, mean)
```

```
> apply(Mar_e, MARGIN = 2, mean)
           X1            X2            X3            X4
 0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
           X5            X6            X7            X8
 0.000000e+00 1.084991e-273 5.154587e-320  0.000000e+00
           X9           X10           X11           X12
 0.000000e+00  0.000000e+00 1.607207e-235 8.474277e-256
          X13           X14           X15           X16
 0.000000e+00  0.000000e+00 3.452718e-295 1.589270e-285
          X17           X18           X19           X20
 0.000000e+00  0.000000e+00 3.051854e-225 1.727217e-254
          X21           X22           X23           X24
 0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
          X25           X26           X27           X28
7.554818e-205  0.000000e+00  0.000000e+00  0.000000e+00
```

```r
# 6
install.packages("mlogit")
install.packages("dfidx")
library(dfidx)
library(mlogit)

quality1 <- quality
colnames(quality1)[1] <- "choice_rev2"
data1 <- left_join(data,quality1,by = "choice_rev2")
# the "choice_rev1" has been changed, so i choose choice_rev2, which won't change the mean

con_choise <- data1$choice_rev1
con_quality <- data1$quality

data2 <-  mlogit.data(data1, varying = 37:282, shape = "wide", sep = '_', choice = "choice_rev1")
# i failed in this step, but i will type the function
```

```r
Con_fun1 <- function(par, choice, quality) {
  choice = con_choice
  quality = con_quality
  n_i = nrow(data)
  n_j = 246 # same with above
  out = mat.or.vec( n_i,n_j )

  out[,1] = 0

  int = par[1:n_j-1]
  qua = par[ (n_j) ]
  for (i in 1:n_i) {
    out[i,] = qua * quality[i]
  }
  for (i in 2:n_j) {
    out[,i] = out[,i] + int[ (i-1) ]
  }
  prob = exp(out)
  prob = sweep(prob, MARGIN=1, FUN="/", STATS=rowSums(prob))
```

```r
  prob_choice = NULL
  for (j in j:n_i){
    prob_choice[j] = prob[j, choice_rev1[j] ]
  }
  prob_choice[prob_choice >0.999999] = 0.999999
  prob_choice[prob_choice <0.000001] = 0.000001
  like = sum( log(prob_choice))
  return(- like)
}
```

```r
 # Marginal effect
prob_fun <- function(par,quality,choice,data){
  n_i = nrow(quality)
  n_j = nrow(n_choice)
  out = mat.or.vec(n_i,n_j)

  par1 = par[1:n_j-1]
  par2 = par[n_j]

  out = par2 *quality[,1]

  for (j in 2:(n_j-1)){
    out = cbind(par2*quality[,j+1]+out,par1[j])
  }

  return(out)
}
```

```r
# 7
# 7.1
# we should choose the second model.
# because in the second model, removing "others" has more influence on school's characteristics rather than individual's.

# 7.2
library(stringr)
out_other <- data %>% filter( str_detect(choice_rev1, "others") == T )
# delete the "other"
# i failed in conditional logit, but i will show the function.

Con_fun1 <- function(par, choice, quality) {
  choice = con_choice
  quality = con_quality
  n_i = nrow(data)
  n_j = 246
  out = mat.or.vec( n_i,n_j )
```

```
  out[,1] = 0

  int = par[1:n_j-1]
  qua = par[ (n_j) ]
  for (i in 1:n_i) {
    out[i,] = qua * quality[i]
  }
  for (i in 2:n_j) {
    out[,i] = out[,i] + int[ (i-1) ]
  }
  prob = exp(out)
  prob = sweep(prob, MARGIN=1, FUN="/", STATS=rowSums(prob))
  prob_choice = NULL
  for (j in j:n_i){
    prob_choice[j] = prob[j, choice_rev1[j] ]
  }
  prob_choice[prob_choice >0.999999] = 0.999999
  prob_choice[prob_choice <0.000001] = 0.000001
  like = sum( log(prob_choice))
  return(- like)
}
```