HW3 Wenxuan Wang

```
setwd("/Users/wenxuan/Desktop/assignment3")
library("data.table")
library("dplyr")
```

#==

Exercise 1:

#==

```
datstu=fread("datstu_v2.csv")
datjss=fread("datjss.csv")
datsss=fread("datsss.csv")
```

```
#Take the longest name for the school name
datsss=datsss %>%
  group_by(schoolcode) %>%
  filter(nchar(schoolname) == max(nchar(schoolname))) %>%
  ungroup()
```

#1.Number of students, schools, programs

The number of students is 340823, the number of schools is 898 (if only calculate the number of schools that students apply, it is 640, because minus 1 NA), the number of programs is 33.

```
> #student
> length(unique(datstu$V1))
[1] 340823
> #340823
>
> #school
> length(unique(datsss$schoolcode))
[1] 898
> #898
> length(unique(c(datstu$schoolcode1,datstu$schoolcode2,datstu$schoolcode3,datstu$s
choolcode4,datstu$schoolcode5,datstu$schoolcode6)))
[1] 641
> #The answer is 641, but when we see the result, we can find 1 NA, thus the number
 of schools is 640
> #640
>
> #program
> length(unique(c(datstu$choicepgm1,datstu$choicepgm2,datstu$choicepgm3,datstu$choi
cepgm4,datstu$choicepgm5,datstu$choicepgm6)))
[1] 33
> #33
```

#2.Number of choices (school, program)

The number of choices is 3086, since I count the NA cases. (If do not count the NA cases, the number is 2773)

```
> #2.Number of choices (school, program)
> choice1=cbind(datstu$schoolcode1,datstu$choicepgm1)
> choice2=cbind(datstu$schoolcode2,datstu$choicepgm2)
> choice3=cbind(datstu$schoolcode3,datstu$choicepgm3)
> choice4=cbind(datstu$schoolcode4,datstu$choicepgm4)
> choice5=cbind(datstu$schoolcode5,datstu$choicepgm5)
> choice6=cbind(datstu$schoolcode6,datstu$choicepgm6)
> all_choice=unique(rbind(choice1,choice2,choice3,choice4,choice5,choice6))
> colnames(all_choice)=c("schoolcode","program")
> length(all_choice)/2
[1] 3086
```

#3.Number of students applying to at least one senior high schools in the same district to home

The number of tudents applying to at least one senior high schools in the same district to home is 262167.

```
#merge the schoolcode and the district
#remove the duplicate values of dataset datsss
datsss_dedup = unique(datsss)
#creat 6 tables with column schoolcode and district
data_district1=cbind(datsss_dedup[,3],datsss_dedup[,4])
data_district2=cbind(datsss_dedup[,3],datsss_dedup[,4])
data_district3=cbind(datsss_dedup[,3],datsss_dedup[,4])
data_district4=cbind(datsss_dedup[,3],datsss_dedup[,4])
data_district5=cbind(datsss_dedup[,3],datsss_dedup[,4])
data_district6=cbind(datsss_dedup[,3],datsss_dedup[,4])

data_district1=unique(data_district1)
data_district2=unique(data_district2)
data_district3=unique(data_district3)
data_district4=unique(data_district4)
data_district5=unique(data_district5)
data_district6=unique(data_district6)

#rename the column in the 6 tables
colnames(data_district1) = c("schoolcode1","sssdistrict1")
colnames(data_district2) = c("schoolcode2","sssdistrict2")
colnames(data_district3) = c("schoolcode3","sssdistrict3")
colnames(data_district4) = c("schoolcode4","sssdistrict4")
colnames(data_district5) = c("schoolcode5","sssdistrict5")
colnames(data_district6) = c("schoolcode6","sssdistrict6")

#merge the schoolcode to find the district of school that each student apply
datstu_district=datstu
datstu_district=merge(datstu_district,data_district1,by="schoolcode1",all.x=TRUE)
datstu_district=merge(datstu_district,data_district2,by="schoolcode2",all.x=TRUE)
datstu_district=merge(datstu_district,data_district3,by="schoolcode3",all.x=TRUE)
datstu_district=merge(datstu_district,data_district4,by="schoolcode4",all.x=TRUE)
datstu_district=merge(datstu_district,data_district5,by="schoolcode5",all.x=TRUE)
datstu_district=merge(datstu_district,data_district6,by="schoolcode6",all.x=TRUE)
```

```
#delete if the  junior high schools is empty
datstu_district=datstu_district%>%filter(datstu_district$jssdistrict!='')
```

```
> b=datstu_district[,17]==datstu_district[,19]|datstu_district[,17]==datstu_district[,20]|datstu
_district[,17]==datstu_district[,21]|datstu_district[,17]==datstu_district[,22]|datstu_district
[,17]==datstu_district[,23]|datstu_district[,17]==datstu_district[,24]
> b=b[complete.cases(b), ]
> summary(b)
   Mode   FALSE    TRUE 
logical   72263  262167 
```

#4.Number of students each senior high school admitted

In increasing order by school code, the table below shows the results.

```
> #4.Number of students each senior high school admitted
> datstu[datstu$rankplace=="1", "school"] = datstu[datstu$rankplace=="1",schoolcode1]
> datstu[datstu$rankplace=="2", "school"] = datstu[datstu$rankplace=="2",schoolcode2]
> datstu[datstu$rankplace=="3", "school"] = datstu[datstu$rankplace=="3",schoolcode3]
> datstu[datstu$rankplace=="4", "school"] = datstu[datstu$rankplace=="4",schoolcode4]
> datstu[datstu$rankplace=="5", "school"] = datstu[datstu$rankplace=="5",schoolcode5]
> datstu[datstu$rankplace=="6", "school"] = datstu[datstu$rankplace=="6",schoolcode6]
> datstu=datstu[complete.cases(datstu$school), ]
> num=datstu%>%group_by(school)%>% summarise(count = n())
> View(num)
```

	school	count
1	10101	398
2	10102	248
3	10103	443
4	10104	220
5	10105	346
6	10106	395
7	10107	306
8	10108	318
9	10109	300
10	10110	535
11	10111	600
12	10112	300
13	10114	350
14	10115	238
15	10116	446

#5.The cutoff of senior high schools (the lowest score to be admitted)

In increasing order by school code, the table below shows the results.

```
> score1 = aggregate(datstu$score, by=list(datstu$school), FUN=min)
> colnames(score1)[1] = 'schoolcode'
> colnames(score1)[2] = 'cutoff'
> View(score1)
```

	schoolcode	cutoff
1	10101	284
2	10102	343
3	10103	316
4	10104	245
5	10105	260
6	10106	293
7	10107	281
8	10108	248
9	10109	257
10	10110	343

#6.The quality of senior high schools (the average score of students admitted)

```
> score2 = aggregate(datstu$score, by=list(datstu$school), FUN=mean)
> colnames(score2)[1] = 'schoolcode'
> colnames(score2)[2] = 'quality'
> View(score2)
```

	schoolcode	quality
1	10101	320.2312
2	10102	394.1492
3	10103	353.8330
4	10104	296.9182
5	10105	351.2139
6	10106	340.1013
7	10107	311.9542
8	10108	303.9025
9	10109	281.8233
10	10110	408.0785

#==

Exercise 2:

#==

#1. the district where the school is located

#2.the latitude and longitude of the district

#3. cutoff (the lowest score to be admitted)

#4.quality (the average score of the students admitted)

#5. size (number of students admitted)

There are 3086 choices in total, so there are 3086 rows of data in the table below.

```
> datsss=datsss[,-1]
> datsss_dedup = unique(datsss)
> data_school=merge(all_choice,datsss_dedup,all.x=TRUE)
> colnames(score1)[1] = 'schoolcode'
> colnames(score1)[2] = 'cutoff'
> colnames(score2)[1] = 'schoolcode'
> colnames(score2)[2] = 'quality'
> colnames(num)[1] = 'schoolcode'
> data_school=merge(data_school,score1,by="schoolcode",all.x=TRUE)
> data_school=merge(data_school,score2,by="schoolcode",all.x=TRUE)
> data_school=merge(data_school,num,by="schoolcode",all.x=TRUE)
> View(data_school)
```

District, latitude, longitude, cutoff, quality, size(count)

Filter

	schoolcode	program	schoolname	sssdistrict	ssslong	ssslat	cutoff	quality	count
1	100101		WA SENIOR HIGH/TECHNICAL SCHOOL, WA	Wa Municipal	-2.2850304	10.030622	198	238.1250	168
2	100101	Home Economics	WA SENIOR HIGH/TECHNICAL SCHOOL, WA	Wa Municipal	-2.2850304	10.030622	198	238.1250	168
3	100101	General Arts	WA SENIOR HIGH/TECHNICAL SCHOOL, WA	Wa Municipal	-2.2850304	10.030622	198	238.1250	168
4	100101	Technical	WA SENIOR HIGH/TECHNICAL SCHOOL, WA	Wa Municipal	-2.2850304	10.030622	198	238.1250	168
5	100102	General Arts	WA SENIOR HIGH SCHOOL, WA	Wa Municipal	-2.2850304	10.030622	250	296.4956	450
6	100102	Home Economics	WA SENIOR HIGH SCHOOL, WA	Wa Municipal	-2.2850304	10.030622	250	296.4956	450
7	100102	Business	WA SENIOR HIGH SCHOOL, WA	Wa Municipal	-2.2850304	10.030622	250	296.4956	450
8	100102	Visual Arts	WA SENIOR HIGH SCHOOL, WA	Wa Municipal	-2.2850304	10.030622	250	296.4956	450
9	100102	Agriculture	WA SENIOR HIGH SCHOOL, WA	Wa Municipal	-2.2850304	10.030622	250	296.4956	450
10	100102	General Science	WA SENIOR HIGH SCHOOL, WA	Wa Municipal	-2.2850304	10.030622	250	296.4956	450
11	100103	General Arts	600103 ST FRANCIS XAVIER JNR SEMINARY, WA	Wa Municipal	-2.2850304	10.030622	NA	NA	NA
12	100103	Business	600103 ST FRANCIS XAVIER JNR SEMINARY, WA	Wa Municipal	-2.2850304	10.030622	NA	NA	NA
13	100103	Agriculture	600103 ST FRANCIS XAVIER JNR SEMINARY, WA	Wa Municipal	-2.2850304	10.030622	NA	NA	NA
14	100104	General Arts	LASSIE-TUOLO SNR SENIOR HIGH. SCHOOL, LASSIE	Wa Municipal	-2.2850304	10.030622	282	326.9333	135
15	100104	General Science	LASSIE-TUOLO SNR SENIOR HIGH. SCHOOL, LASSIE	Wa Municipal	-2.2850304	10.030622	282	326.9333	135
16	100104	Home Economics	LASSIE-TUOLO SNR SENIOR HIGH. SCHOOL, LASSIE	Wa Municipal	-2.2850304	10.030622	282	326.9333	135

Showing 1 to 15 of 3,086 entries, 9 total columns

#===

Exercise 3:

#===

#1.Using the formula, where ssslong and ssslat are the coordinates of the district of the school (students apply to), while jsslong and jsslat are the coordinates of the junior high school, calculate the distance between junior high school, and senior high school. You should generate a value of distance for each of students' choices.

#First, remove the duplicate values of dataset datsss

```
> datsss_dedup = unique(datsss)
```

#Second, create 7 databases containing longitude, latitude, or district

```
> merge1=cbind(datsss_dedup$schoolcode,datsss_dedup$ssslong,datsss_dedup$ssslat)
> merge2=cbind(datsss_dedup$schoolcode,datsss_dedup$ssslong,datsss_dedup$ssslat)
> merge3=cbind(datsss_dedup$schoolcode,datsss_dedup$ssslong,datsss_dedup$ssslat)
> merge4=cbind(datsss_dedup$schoolcode,datsss_dedup$ssslong,datsss_dedup$ssslat)
> merge5=cbind(datsss_dedup$schoolcode,datsss_dedup$ssslong,datsss_dedup$ssslat)
> merge6=cbind(datsss_dedup$schoolcode,datsss_dedup$ssslong,datsss_dedup$ssslat)
> merge7=cbind(datsss_dedup$ssslong,datsss_dedup$ssslat,datsss_dedup$sssdistrict)
> merge7=as.data.frame(merge7)
> merge7=unique(merge7)
> merge7=na.omit(merge7)
```

#rename the column

```
> colnames(merge1) = c("schoolcode1","ssslong1","ssslat1")
> colnames(merge2) = c("schoolcode2","ssslong2","ssslat2")
> colnames(merge3) = c("schoolcode3","ssslong3","ssslat3")
> colnames(merge4) = c("schoolcode4","ssslong4","ssslat4")
> colnames(merge5) = c("schoolcode5","ssslong5","ssslat5")
> colnames(merge6) = c("schoolcode6","ssslong6","ssslat6")
> colnames(merge7) = c("jsslong","jsslat","jssdistrict")
```

#merge all the 7 dataset with datstu

```
> merge=merge(datstu,merge1,by="schoolcode1",all.x=TRUE)
> merge=merge(merge,merge2,by="schoolcode2",all.x=TRUE)
> merge=merge(merge,merge3,by="schoolcode3",all.x=TRUE)
> merge=merge(merge,merge4,by="schoolcode4",all.x=TRUE)
> merge=merge(merge,merge5,by="schoolcode5",all.x=TRUE)
> merge=merge(merge,merge6,by="schoolcode6",all.x=TRUE)
> merge=merge(merge,merge7,by="jssdistrict",all.x=TRUE)
> merge$jsslong = as.numeric(merge$jsslong)
> merge$jsslat=as.numeric(merge$jsslat)
```

#we can calculate the distance with NA or without NA, now I get the result with NA

```
> merge$distance1= sqrt((69.172*(merge$ssslong1-merge$jsslong)*cos((merge$jsslat/57.3))^2+(69.172*(merge
$ssslat1-merge$jsslat))^2))

> merge$distance2= sqrt((69.172*(merge$ssslong2-merge$jsslong)*cos((merge$jsslat/57.3))^2+(69.172*(merge
$ssslat2-merge$jsslat))^2))

> merge$distance3= sqrt((69.172*(merge$ssslong3-merge$jsslong)*cos((merge$jsslat/57.3))^2+(69.172*(merge
$ssslat3-merge$jsslat))^2))

> merge$distance4= sqrt((69.172*(merge$ssslong4-merge$jsslong)*cos((merge$jsslat/57.3))^2+(69.172*(merge
$ssslat4-merge$jsslat))^2))

> merge$distance5= sqrt((69.172*(merge$ssslong5-merge$jsslong)*cos((merge$jsslat/57.3))^2+(69.172*(merge
$ssslat5-merge$jsslat))^2))

> merge$distance6= sqrt((69.172*(merge$ssslong6-merge$jsslong)*cos((merge$jsslat/57.3))^2+(69.172*(merge
$ssslat6-merge$jsslat))^2))
```

I calculated the distance from each student's choice to their junior school. The results are as follows.

g6	ssslat6	jssslong	jssslat	distance1	distance2	distance3	distance4	distance5	distance6
13774	11.036352	−0.18813774	11.036352	406.8263	2.915029	19.161503	0.000000	31.976754	0.000000
36101	6.258390	−1.99056137	4.834704	405.1130	359.383876	395.018871	397.033328	17.767689	98.253233
55472	5.081101	−1.62365472	5.081101	404.2442	391.672387	391.672387	391.672387	0.000000	0.000000
30215	5.141226	−2.80322027	5.069508	404.0933	412.955938	0.000000	0.000000	7.637895	7.637895
49449	10.557073	−1.21744096	10.909423	403.1230	403.123050	223.442102	235.389735	5.449285	24.207953
13774	11.036352	−0.78770429	10.924120	400.7568	404.104307	10.012653	0.000000	24.656783	10.012653
24253	5.617353	−1.30659390	5.153656	398.1456	398.145588	151.326114	151.326114	0.000000	32.659275
30215	5.141226	−2.31180215	5.141226	392.8704	407.956315	407.956315	392.870392	8.032123	0.000000
30215	5.141226	−2.80322027	5.069508	392.5773	392.577345	7.637895	7.637895	7.637895	7.637895
30215	5.141226	−1.62365472	5.081101	391.6724	403.191225	391.672387	297.691993	0.000000	NaN
55472	5.081101	−1.62365472	5.081101	391.6724	403.191225	391.672387	403.191225	0.000000	0.000000
35322	5.276049	−1.62365472	5.081101	391.6724	391.672387	391.672387	378.809804	0.000000	12.521173
55472	5.081101	−1.62365472	5.081101	391.6724	391.672387	391.672387	404.244245	0.000000	0.000000
55472	5.081101	−1.62365472	5.081101	391.6724	391.672387	0.000000	0.000000	0.000000	0.000000
55472	5.081101	−1.62365472	5.081101	391.6724	391.672387	0.000000	0.000000	0.000000	0.000000

Showing 1 to 14 of 139,224 entries, 39 total columns

#===

Exercise 4:

#===

#1.Recode the schoolcode into its first three digits (substr). Call this new variable scode rev.

```
> all_choice_new=data.frame(all_choice)
> all_choice_new$scode_rev=substring(all_choice_new$schoolcode,1,3)
```

	schoolcode	program	scode_rev
1	50112	Home Economics	501
2	70102	General Arts	701
3	50702	Business	507
4	90501	Visual Arts	905
5	51802	Home Economics	518
6	10102	General Arts	101
7	80301	General Arts	803
8	40301	General Arts	403
9	21303	Business	213
10	80101	General Arts	801
11	100201	General Science	100
12	30603	Business	306
13	80101	Business	801
14	90301	Technical	903
15	40903	General Arts	409

Showing 1 to 15 of 3,086 entries, 3 total columns

#2.Recode the program variable into 4 categories: arts (general arts and visual arts), economics (business,and home economics), science (general science) and others. Call this new variable pgm rev.

```
> all_choice_new$pgm_rev[all_choice_new$program == "General Arts"] = "arts"
> all_choice_new$pgm_rev[all_choice_new$program == "Visual Arts"] = "arts"
> all_choice_new$pgm_rev[all_choice_new$program == "Business"] = "economics"
> all_choice_new$pgm_rev[all_choice_new$program == "Home Economics"] = "economics"
> all_choice_new$pgm_rev[all_choice_new$program == "General Science"] = "science"
> all_choice_new$pgm_rev[is.na(all_choice_new$pgm_rev) ==TRUE] = "others"
> all_choice_new
```

	schoolcode	program	scode_rev	pgm_rev
1	50112	Home Economics	501	economics
2	70102	General Arts	701	arts
3	50702	Business	507	economics
4	90501	Visual Arts	905	arts
5	51802	Home Economics	518	economics
6	10102	General Arts	101	arts
7	80301	General Arts	803	arts
8	40301	General Arts	403	arts
9	21303	Business	213	economics
10	80101	General Arts	801	arts
11	100201	General Science	100	science

Showing 1 to 11 of 3,086 entries, 4 total columns

#3.Create a new choice variable choice_rev.

	schoolcode	program	scode_rev	pgm_rev	choice_rev
1	50112	Home Economics	501	economics	501 economics
2	70102	General Arts	701	arts	701 arts
3	50702	Business	507	economics	507 economics
4	90501	Visual Arts	905	arts	905 arts
5	51802	Home Economics	518	economics	518 economics
6	10102	General Arts	101	arts	101 arts
7	80301	General Arts	803	arts	803 arts
8	40301	General Arts	403	arts	403 arts
9	21303	Business	213	economics	213 economics
10	80101	General Arts	801	arts	801 arts
11	100201	General Science	100	science	100 science

Showing 1 to 11 of 3,086 entries, 5 total columns

#4.Recalculate the cutoff and the quality for each recoded choice.

The results are in the following two tables.

#admitted school

```
datstu[datstu$rankplace=="1", "school"] = datstu[datstu$rankplace=="1",schoolcode1]
datstu[datstu$rankplace=="2", "school"] = datstu[datstu$rankplace=="2",schoolcode2]
datstu[datstu$rankplace=="3", "school"] = datstu[datstu$rankplace=="3",schoolcode3]
datstu[datstu$rankplace=="4", "school"] = datstu[datstu$rankplace=="4",schoolcode4]
datstu[datstu$rankplace=="5", "school"] = datstu[datstu$rankplace=="5",schoolcode5]
datstu[datstu$rankplace=="6", "school"] = datstu[datstu$rankplace=="6",schoolcode6]
datstu=datstu[complete.cases(datstu$school), ]
datstu$school_new=substring(datstu$school,1,3)
```

#admitted program

```
> datstu[datstu$rankplace=="1", "program"] = datstu[datstu$rankplace=="1",choicepgm1]
> datstu[datstu$rankplace=="2", "program"] = datstu[datstu$rankplace=="2",choicepgm2]
> datstu[datstu$rankplace=="3", "program"] = datstu[datstu$rankplace=="3",choicepgm3]
> datstu[datstu$rankplace=="4", "program"] = datstu[datstu$rankplace=="4",choicepgm4]
> datstu[datstu$rankplace=="5", "program"] = datstu[datstu$rankplace=="5",choicepgm5]
> datstu[datstu$rankplace=="6", "program"] = datstu[datstu$rankplace=="6",choicepgm6]
> datstu=datstu[complete.cases(datstu$program), ]
```

#program categories

```
> datstu$program_new[datstu$program == "General Arts"] = "arts"
> datstu$program_new[datstu$program == "Visual Arts"] = "arts"
> datstu$program_new[datstu$program == "Business"] = "economics"
> datstu$program_new[datstu$program == "Home Economics"] = "economics"
> datstu$program_new[datstu$program == "General Science"] = "science"
> datstu$program_new[is.na(datstu$program_new) ==TRUE] = "others"
> #Create a new choice variable choice_rev
> datstu$choice_rev=paste(datstu$school_new,datstu$program_new)
```

#The cutoff of each recoded choice (the lowest score to be admitted)

```
> score3 = aggregate(datstu$score, by=list(datstu$choice_rev), FUN=min)
> colnames(score3)=c("choice_rev","cutoff")
```

	choice_rev	cutoff
1	100 arts	194
2	100 economics	195
3	100 others	191
4	100 science	228
5	101 arts	243
6	101 economics	205
7	101 others	257
8	101 science	203
9	102 arts	216
10	102 economics	206
11	102 others	209

Showing 1 to 12 of 425 entries, 2 total columns

#The quality of each recoded choice (the average score of students admitted)

```
> score4 = aggregate(datstu$score, by=list(datstu$choice_rev), FUN=mean)
> colnames(score4)=c("choice_rev","quality")
> datstu1=cbind(datstu[,23],datstu[,2])
> datstu11=head(datstu1[order(datstu1$score,decreasing = TRUE), ], 20000)
```

	choice_rev	quality
1	100 arts	275.5233
2	100 economics	264.4993
3	100 others	245.6381
4	100 science	305.1814
5	101 arts	340.0850
6	101 economics	326.3979
7	101 others	313.2753
8	101 science	368.7612
9	102 arts	315.5544
10	102 economics	308.9986
11	102 others	280.9509

Showing 1 to 12 of 425 entries, 2 total columns

#5.Consider the 20,000 highest score students.

```
#5.Consider the 20,000 highest score students.
datstu=merge(datstu,score4,by="choice_rev",all.x=TRUE)
datstu_20000=head(datstu[order(datstu$score,decreasing = TRUE), ], 20000)
datstu_20000=cbind(datstu_20000[,1],datstu_20000[,3],datstu_20000[,24])
```

	choice_rev	score	quality
2123	100 arts	416	275.5233
2514	100 arts	413	275.5233
5396	100 arts	396	275.5233
7017	100 arts	389	275.5233
7574	100 arts	387	275.5233
8681	100 arts	383	275.5233
9610	100 arts	380	275.5233
9920	100 arts	379	275.5233
9921	100 arts	379	275.5233
11260	100 arts	375	275.5233
11261	100 arts	375	275.5233
11606	100 arts	374	275.5233
11971	100 arts	373	275.5233
12333	100 arts	372	275.5233

Showing 1 to 14 of 20,000 entries, 3 total columns

In my 20,000 dataset, there are 271 choices. Thus, the following questions base on this

dataset.

```
> unique(datstu_20000$choice_rev)
  [1] "301 science"   "210 science"   "501 science"   "101 science"   "301 arts"      "301 economics"
  [7] "101 arts"      "203 science"   "100 science"   "301 others"    "701 science"   "501 economics"
 [13] "201 economics" "201 science"   "211 science"   "203 arts"      "210 economics" "210 others"
 [19] "501 arts"      "101 economics" "401 science"   "203 economics" "211 arts"      "101 others"
 [25] "201 arts"      "303 science"   "502 arts"      "201 others"    "210 arts"      "206 economics"
 [31] "502 science"   "401 others"    "102 economics" "503 arts"      "401 economics" "102 arts"
 [37] "601 science"   "701 arts"      "211 economics" "303 arts"      "505 arts"      "705 science"
 [43] "303 economics" "502 economics" "204 arts"      "401 arts"      "706 science"   "213 arts"
 [49] "213 science"   "701 economics" "102 science"   "801 science"   "304 arts"      "705 economics"
 [55] "213 economics" "305 science"   "403 arts"      "102 others"    "100 arts"      "105 others"
 [61] "305 economics" "306 others"    "306 science"   "505 economics" "601 economics" "707 economics"
 [67] "104 arts"      "303 others"    "502 others"    "902 others"    "505 science"   "601 arts"
 [73] "215 arts"      "517 economics" "904 science"   "204 economics" "203 others"    "308 science"
 [79] "309 arts"      "517 arts"      "202 others"    "204 science"   "309 science"   "505 others"
 [85] "705 arts"      "304 science"   "503 economics" "605 science"   "612 arts"      "801 economics"
 [91] "304 economics" "309 economics" "512 science"   "706 arts"      "105 economics" "306 arts"
 [97] "506 science"   "105 arts"      "215 science"   "601 others"    "309 others"    "503 science"
[103] "512 arts"      "612 science"   "105 science"   "310 science"   "706 others"    "905 arts"
[109] "905 science"   "306 economics" "409 economics" "501 others"    "603 arts"      "901 others"
[115] "405 science"   "215 economics" "216 economics" "104 science"   "204 others"    "206 arts"
[121] "710 science"   "907 others"    "103 others"    "103 science"   "202 arts"      "706 economics"
[127] "710 economics" "208 science"   "506 arts"      "904 arts"      "100 economics" "202 science"
[133] "205 science"   "606 science"   "701 others"    "712 economics" "801 others"    "904 economics"
[139] "311 economics" "407 others"    "510 economics" "610 economics" "710 arts"      "802 science"
[145] "304 others"    "307 others"    "308 arts"      "402 arts"      "503 others"    "510 arts"

[151] "801 arts"      "517 science"   "205 economics" "607 science"   "707 science"   "104 economics"
[157] "310 arts"      "612 economics" "709 arts"      "510 science"   "603 science"   "104 others"
[163] "409 science"   "512 economics" "202 economics" "208 economics" "506 economics" "702 arts"
[169] "310 economics" "409 arts"      "608 others"    "302 science"   "311 arts"      "602 science"
[175] "305 arts"      "905 others"    "208 arts"      "312 arts"      "402 science"   "404 arts"
[181] "602 arts"      "904 others"    "312 others"    "407 arts"      "510 others"    "518 science"
[187] "705 others"    "516 economics" "807 science"   "205 others"    "407 economics" "605 arts"
[193] "606 arts"      "707 arts"      "206 science"   "404 others"    "512 others"    "516 arts"
[199] "709 economics" "406 economics" "410 arts"      "518 economics" "703 science"   "103 economics"
[205] "403 economics" "511 science"   "205 arts"      "213 others"    "308 economics" "405 arts"
[211] "507 arts"      "508 arts"      "603 economics" "207 arts"      "311 science"   "312 economics"
[217] "312 science"   "508 others"    "607 arts"      "709 science"   "514 arts"      "216 arts"
[223] "402 others"    "508 science"   "518 arts"      "708 economics" "103 arts"      "406 science"
[229] "514 science"   "206 others"    "214 science"   "215 others"    "307 arts"      "403 science"
[235] "516 science"   "310 others"    "507 science"   "603 others"    "605 economics" "712 science"
[241] "517 others"    "602 economics" "605 others"    "609 arts"      "207 others"    "209 science"
[247] "516 others"    "610 science"   "702 economics" "712 arts"      "704 others"    "405 others"
[253] "607 economics" "704 arts"      "903 others"    "212 arts"      "402 economics" "602 others"
[259] "610 arts"      "711 arts"      "209 arts"      "209 others"    "207 science"   "513 arts"
[265] "609 science"   "711 others"    "212 science"   "214 economics" "507 economics" "803 arts"
[271] "901 arts"
```

#==

Exercise 5:

#==

#1.Propose a model specification. Write the Likelihood function.

#Consider the 20,000 highest score students.

```
datstu111=cbind(datstu_20000[,1],datstu_20000[,2])
datstu2 = as.data.frame(datstu111)
class(datstu2)
like_fun = function(param,dat_stu)
{
  score =  as.vector(dat_stu$score)
  cats = as.vector(sort(unique(dat_stu[,1])))
  ch = as.vector(match(dat_stu[,1], cats))
  ni = nrow(dat_stu)
  nj = length(cats)
  ut = mat.or.vec(ni,nj)
  # multinomial logit
  pn1    = param[1:nj]
  pn2    = param[(nj+1):(2*nj)]
  for (j in 1:nj)
  {
    ut[,j] = score*pn1[j] + pn2[j]
  }
  prob   = exp(ut)            # exp(XB)
  #sprob  = rowsums(prob)      # sum_j exp(XB) denominator
  prob   = sweep(prob,MARGIN=1,FUN="/",STATS=rowSums(prob))
  # match prob to actual choices
  probc = NULL
  for (i in 1:ni)
  {
    probc[i] = prob[i,ch[i]]
  }
  probc[probc>0.999999] = 0.999999
  probc[probc<0.000001] = 0.000001
  like = sum(log(probc)) / ni
  return(-like)
}
nj = length(as.vector(unique(datstu2[,1])))
print(nj)
res1 = optim(runif(2 * nj,min=-0.1,max=0),fn=like_fun,
             method="BFGS",control=list(trace=5,REPORT=1,maxit=10),
             dat_stu=datstu2,hessian=FALSE)
```

```
> res1 = optim(runif(2 * nj,min=-0.1,max=0),fn=like_fun,
+              method="BFGS",control=list(trace=5,REPORT=1,maxit=10),
+              dat_stu=datstu2,hessian=FALSE)
initial  value 12.082575
iter   2 value 11.813250
iter   3 value 11.039611
iter   4 value 10.936658
iter   5 value 10.872245
iter   6 value 10.716199
iter   7 value 10.703339
iter   8 value 10.677898
iter   9 value 10.613959
iter  10 value 10.592330
final  value 10.592330
```

#2.Estimate parameters and compute the marginal effect of the proposed model.

The parameters for the 271 choices are as follows. Since each choice has two parameters, there are 542 values in total.

```
> parameters=res1$par
> parameters
  [1] -0.0673487359 -0.0610494059 -0.0941333388 -0.0867712282 -0.0540551738 -0.0626880465  0.0116935413
  [8] -0.0460481435 -0.0539855133 -0.0459203887 -0.0371075148 -0.0576345878 -0.0262264023 -0.0178024307
 [15] -0.0614545397 -0.0500101167 -0.0716370029 -0.0682894722 -0.0291537348 -0.0367044529 -0.0637654101
 [22] -0.0127370127 -0.0385937569 -0.0622706652 -0.0782298685 -0.0508564033 -0.0149111031 -0.0975880065
 [29] -0.0274098751 -0.0879137622 -0.0113934259 -0.0626040742 -0.0982523184 -0.0310093913 -0.0545377178
 [36] -0.0036185942 -0.0877355496 -0.0365715694 -0.0583881169 -0.0446035422 -0.0525904956 -0.0111611308
 [43] -0.0859164396 -0.0704716630 -0.0628621177 -0.0704371352 -0.0229633245 -0.0971967720 -0.0881025372
 [50] -0.0865348173 -0.0408274166 -0.0823133239 -0.0326217522 -0.0908028277 -0.0143277028 -0.0259182782
 [57] -0.0253600001 -0.0423209905 -0.0493664983 -0.0005503467 -0.0688804908 -0.0981647289 -0.0676423314
 [64] -0.0668922330 -0.0123120199 -0.0317614710  0.0087140086 -0.0234002354 -0.0759324567 -0.0332368072
 [71] -0.0439678667 -0.0441145307 -0.0077226219 -0.0181283188 -0.0538231223 -0.0658709406 -0.0936935505
 [78] -0.0874390538 -0.0876117755 -0.0758704114 -0.0657987345 -0.0534824826 -0.0145135860 -0.0704651033
 [85] -0.0386718473 -0.0071689248 -0.0486861500 -0.0926720374 -0.0799728442 -0.0673741299 -0.0734259279
 [92] -0.0494582565 -0.0794661517  0.0097871235 -0.0982119980 -0.0352114441 -0.0471540265 -0.0257526421
 [99] -0.0998269790 -0.0220299762 -0.0616555044 -0.0300988117 -0.0294746521 -0.0487852123 -0.0608280082
[106] -0.0633848196 -0.0188819153 -0.0526461420 -0.0169409629 -0.0876156174 -0.0401368803 -0.0483942428
[113] -0.0142182173 -0.0949519219 -0.0506721764 -0.0795340519 -0.0636935760  0.0146459444  0.0020074299
[120] -0.0576703093 -0.0555329480 -0.0266498033 -0.0534875521 -0.0634418568 -0.0973695515 -0.0661819472
[127] -0.0989360098 -0.0684613354 -0.0189275476 -0.0221288121 -0.0477746077 -0.0842857936 -0.0493905568
[134] -0.0656020499 -0.0446515603 -0.0159794269 -0.0620094416 -0.0300550240 -0.0789043417 -0.0784993494
[141] -0.0542988371 -0.0530406283  0.0076566375  0.0200290890 -0.0496750098  0.0135207431 -0.0563832548
[148] -0.0494716402 -0.0479302191  0.0000946851 -0.0160903651 -0.0379324217 -0.0305436837 -0.0250921831
[155] -0.0666745910 -0.0624640473 -0.0813813142 -0.0019787831 -0.0123290447 -0.0972764386 -0.0178581367
[162] -0.0961232320 -0.0677943734 -0.0144455514 -0.0440473533 -0.0330713903 -0.0764528277 -0.0423265846
[169] -0.0044245488 -0.0296048697  0.0005082929 -0.0854071447 -0.0440076273 -0.0448875998 -0.0725987291
[176] -0.0885238008 -0.0657292329 -0.0752798273 -0.0223448296 -0.0190862948 -0.0115497974 -0.0479621952
[183] -0.0990992153 -0.0078619101 -0.0459718490 -0.0341218372 -0.0556148722 -0.0162773234 -0.0358544910
[190] -0.0967871290 -0.0456548898 -0.0046574666 -0.0746173404 -0.0542206780 -0.0957744537 -0.0398632983

[197] -0.0955580774 -0.0252502054 -0.0282602103 -0.0923945167 -0.0802244727 -0.0484956578 -0.0669189382
[204] -0.0454569158 -0.0872751980 -0.0680401344 -0.0969562138 -0.0403668521 -0.0400266410 -0.0287999087
[211] -0.0787728656 -0.0114812638 -0.0987366414 -0.0500139938 -0.0432253071 -0.0413988181 -0.0792533564
[218] -0.0341627245 -0.0616683780 -0.0260565481  0.0133815014 -0.0474277474 -0.0210051741 -0.0415221506
[225] -0.0111950695 -0.0458178619 -0.0447232762 -0.0155609025 -0.0315872080 -0.0552069794 -0.0256501025
[232] -0.0136844144 -0.0958466320 -0.0388164452 -0.0909614975 -0.0358563344 -0.0706026081 -0.0905625720
[239] -0.0218396515 -0.0834747359 -0.0688953485 -0.0922691570 -0.0817428925 -0.0133465788 -0.0292676559
[246] -0.0115581102 -0.0176993743 -0.0391327671 -0.0886233309 -0.0463170370 -0.0449924490 -0.0639909563
[253] -0.0424100911 -0.0868973879 -0.0564582584 -0.0089680431 -0.0464061721 -0.0964081271 -0.0676077612
[260] -0.0118082828 -0.0650800097 -0.0528623717 -0.0331903220 -0.0737977172 -0.0613254872 -0.0140915388
[267] -0.0770923601 -0.0771707403 -0.0233640498 -0.0849082486 -0.0413505181 -0.0515846545 -0.0793254091
[274] -0.0769870646 -0.0980367313 -0.0719795345 -0.0643895055 -0.0579326064 -0.0343417464 -0.0338144886
[281] -0.0845177462 -0.0999231381 -0.0453006784 -0.0861645825 -0.0541487467 -0.0121884811 -0.0670867751
[288] -0.0746168253 -0.0184808260 -0.0658179292 -0.0769916398 -0.0051943888 -0.0719366203 -0.0873227432
[295] -0.0513318422 -0.0606583546 -0.0466053389 -0.0123889746 -0.0649400662 -0.0137570149 -0.0071572688
[302] -0.0383286949 -0.0227074951 -0.0112988522 -0.0032307079 -0.0104231072 -0.0762693935 -0.0976675983
[309] -0.0811047102 -0.0721071335 -0.0261038718 -0.0235266820 -0.0392835808 -0.0033200841 -0.0049746047
[316] -0.0417552516 -0.0115136326 -0.0983688012 -0.0401533082 -0.0498589349 -0.0183264436 -0.0313599718
[323] -0.0868780178 -0.0393513728 -0.0010782868 -0.0202006251 -0.0493671657 -0.0994679920 -0.0310536104
[330] -0.0529867694 -0.0624951248 -0.0004884171 -0.0039318000 -0.0586572185 -0.0459845900 -0.0976650715
[337] -0.0524208815 -0.0451256720 -0.0380371329 -0.0334201173 -0.0683544553 -0.0062347256 -0.0535951102
[344] -0.0553636339 -0.0009415089 -0.0532125873 -0.0417557465 -0.0732301075 -0.0732914212 -0.0894896941
[351] -0.0538387923 -0.0176100430 -0.0671343999 -0.0244803156 -0.0168617085 -0.0500655225 -0.0856738380
[358] -0.0424335016 -0.0598543603 -0.0366774196 -0.0666034145 -0.0857248053 -0.0723566209 -0.0897179837
[365] -0.0167593376 -0.0458007162 -0.0908566800 -0.0267755609 -0.0193172709 -0.0756533507 -0.0968175187
[372] -0.0841330892 -0.0977389340 -0.0128399633 -0.0893413363 -0.0146218520 -0.0948751743 -0.0685945831
[379] -0.0735398079 -0.0616892715 -0.0476089394 -0.0026472842 -0.0355242127 -0.0991211153 -0.0060626745
[386] -0.0565793824 -0.0722912691 -0.0316485978 -0.0799926447 -0.0688326466 -0.0718160755 -0.0318142579
```

```
[393] -0.0323938399 -0.0097256859 -0.0059901451 -0.0768771352 -0.0962104447 -0.0218887002 -0.0679485668
[400] -0.0072532531 -0.0361490475 -0.0846080829 -0.0313757987 -0.0033506960 -0.0193064986 -0.0177584721
[407] -0.0045930199 -0.0616921503 -0.0484815080 -0.0039520151 -0.0555287827 -0.0451732355 -0.0263313401
[414] -0.0892135754 -0.0804467211 -0.0686659718 -0.0805324976 -0.0946948422 -0.0222618720 -0.0739176230
[421] -0.0898414147 -0.0277043802 -0.0126264309 -0.0149052667 -0.0063168084 -0.0031518379 -0.0783976349
[428] -0.0862250960 -0.0675239317 -0.0935820522 -0.0185186857 -0.0962835986 -0.0731025613 -0.0772675060
[435] -0.0062370467 -0.0169767821 -0.0937810171 -0.0587469894 -0.0453915520 -0.0303024354 -0.0447620061
[442] -0.0702972549 -0.0157607270 -0.0198522335 -0.0958720721 -0.0863163730 -0.0851267069 -0.0243461079
[449] -0.0475746411 -0.0655376317 -0.0037778211 -0.0570783638 -0.0979241084 -0.0531769584 -0.0902317975
[456] -0.0563996033 -0.0042959957 -0.0032439004 -0.0475973951 -0.0190290886 -0.0886057209 -0.0002773821
[463] -0.0695405997 -0.0286494612 -0.0616016754 -0.0502639496 -0.0048479726 -0.0103384244 -0.0896899550
[470] -0.0238686845 -0.0247862298 -0.0525012235 -0.0768362517 -0.0959159885 -0.0398802199 -0.0383486166
[477] -0.0486514860 -0.0459625336 -0.0691179177 -0.0886185238 -0.0844323383 -0.0294305703 -0.0519133006
[484] -0.0159954661 -0.0246717307 -0.0696447628 -0.0917745093 -0.0335507663 -0.0828294465 -0.0298657450
[491] -0.0096599948 -0.0236395335 -0.0420734731 -0.0932187589 -0.0602649215 -0.0483005613 -0.0678418515
[498] -0.0116694559 -0.0740127462 -0.0633053279 -0.0930412656 -0.0161179624 -0.0996995638 -0.0619606041
[505] -0.0540916142 -0.0237879312 -0.0927170344 -0.0573613384 -0.0412770750 -0.0257665873 -0.0933948279
[512] -0.0189343042 -0.0021121194 -0.0131299222 -0.0633712027 -0.0970576283 -0.0453826185 -0.0316463719
[519] -0.0214549475 -0.0800354174 -0.0092745673 -0.0567243584 -0.0746245853 -0.0369304377 -0.0265963951
[526] -0.0123322272 -0.0959334981 -0.0713729847 -0.0306321224 -0.0687240225 -0.0024535016 -0.0763156079
[533] -0.0771462510 -0.0323415560 -0.0890773304 -0.0662542233 -0.0301714930 -0.0311782986 -0.0495708376
[540] -0.0803347371 -0.0693891768 -0.0996991365
```

#marginal effect

```
#marginal effect
#after we get the parameters, write it in the matrix way
beta = mat.or.vec(2,nj)
beta[1,] = res1$par[1:271]
beta[2,] = res1$par[272:542]
# build a matrix with columns "1" and scores
X=cbind(rep(1,20000),datstu2$score)
#creat a matrix with 20000 row and 271 columns
a=mat.or.vec(20000,271)
#the sum of the exponential function
exponential=exp(X%*%beta[,2:271])
rowSums(exponential)
sum=as.matrix(rowSums(exponential))
a=mat.or.vec(20000,271)
#give value for the matrix a
a[,1]=1/(1+sum)
a[,2]=exp(X %*% beta[,2])/(1+sum)
```

```
for (i in 1:270){
  value=exp(X %*% beta[,i+1])/(1+sum)
  a[,i+1]=value
}

#creat a 20000*271 matrix with the first column 0
b=mat.or.vec(20000,271)
bbeta=c(res1$par[272:542])
#marginal effect:{exp(X *beta[,i+1])/(1+sum)}*(par-sum beta)
for (i in 1:20000) {
  b[i,]=a[i,]*(bbeta-sum(a[i,]*bbeta))
}

marginal_effect1=apply(b, 2, mean)
marginal_effect1
```

The marginal effect for the 271 choices are shown in the table.

```
> marginal_effect1
  [1] -2.475316e-04 -1.299731e-10 -2.447800e-08 -6.198722e-13 -3.932208e-06
  [6] -9.775683e-17 -7.344913e-06 -2.298846e-06  1.713979e-04 -2.291985e-16
 [11] -4.208553e-07 -4.608424e-08 -1.148270e-14 -1.601422e-13 -1.788515e-18
 [16] -5.562250e-14 -1.044362e-15 -1.750680e-16 -4.392850e-17 -9.290578e-12
 [21] -2.135326e-10 -1.294279e-11 -1.288201e-08 -6.962067e-07 -4.451176e-14
 [26] -2.578432e-08 -5.195592e-09 -7.491216e-09 -6.970647e-10 -4.135472e-18
 [31] -1.050330e-15 -1.550828e-09 -1.242966e-16 -8.909028e-17 -2.479646e-09
 [36] -5.875072e-12 -7.304511e-10 -2.431645e-16 -3.333929e-08 -1.352258e-12
 [41] -1.237716e-12 -4.756384e-07 -2.865768e-07 -4.996766e-11 -6.079953e-17
 [46] -5.958028e-16 -1.323728e-15 -3.233402e-09 -3.025508e-07 -7.202714e-12
 [51] -7.505377e-14 -4.938700e-09 -2.109296e-14 -1.685424e-17 -5.268032e-18
 [56] -5.637591e-06 -1.791666e-06 -9.341251e-13 -5.294189e-16 -7.023491e-12
 [61] -1.875315e-15 -1.330915e-14 -3.224382e-05 -1.416076e-05 -1.437149e-16
 [66] -5.407151e-06 -2.794215e-16 -1.698409e-09 -4.415945e-16 -5.546916e-18
 [71] -2.953350e-12 -6.840331e-07 -2.264467e-10 -2.098747e-08 -4.084903e-17
 [76] -8.507010e-08 -1.386556e-14  1.415534e-05 -1.358244e-18 -3.676970e-18
 [81] -5.065571e-14 -4.813010e-09 -1.629487e-18 -8.735948e-06 -2.404578e-18
 [86] -2.492030e-17 -2.884100e-14 -3.398070e-05  6.453888e-05 -7.184064e-06
 [91] -1.840262e-07 -1.439967e-06 -1.042405e-10 -3.542597e-15 -3.321677e-05
 [96] -4.435462e-07 -1.152323e-10  3.399785e-04 -8.842774e-11 -4.199234e-13
[101] -4.396602e-13 -3.401794e-18 -1.237848e-06 -5.937919e-06 -1.065467e-14
[106] -3.314998e-05 -5.519705e-10 -5.016393e-08 -6.771211e-13 -5.764531e-11
[111] -7.359110e-12 -2.091235e-05 -3.823114e-13 -1.549108e-16 -1.908875e-12
[116] -2.191800e-16 -1.169576e-08 -4.485355e-08 -1.153512e-06 -2.930964e-09
[121] -3.353589e-05 -4.128009e-17 -7.792736e-16 -3.460852e-05 -6.176676e-18
[126] -3.292251e-16 -1.826274e-07 -4.764276e-13 -6.460097e-17 -1.216498e-06
[131] -6.299075e-11 -6.876089e-12 -2.799738e-11 -5.757707e-15 -3.510118e-12
[136] -1.225200e-11 -4.007724e-16 -4.139455e-07 -2.869074e-05 -2.281224e-05
[141] -7.869824e-15 -2.654794e-14 -1.250398e-05 -5.572396e-06  3.050879e-04

[146] -1.230833e-07 -1.017087e-17 -1.260885e-18 -2.840918e-11 -1.595221e-12
[151] -4.905632e-12 -3.663997e-10 -9.527675e-10 -3.688055e-15  6.210951e-05
[156] -3.056339e-05 -1.193597e-09 -6.093433e-09 -3.555952e-05 -1.799450e-17
[161] -1.599485e-12 -1.631412e-05 -3.145357e-05 -6.095606e-14 -3.151842e-09
[166] -1.242616e-08 -1.170725e-12 -2.799314e-13 -4.925388e-11 -9.616815e-06
[171] -4.568759e-10 -7.839456e-07 -1.006688e-10 -2.861522e-05 -9.924815e-11
[176] -1.634681e-08 -4.373654e-15 -2.633891e-05 -1.913819e-06 -2.571867e-16
[181] -1.323738e-13 -2.728091e-05 -3.483718e-05 -3.284697e-13 -8.983190e-18
[186] -2.343248e-16 -7.167214e-11 -2.104064e-11 -9.726324e-15 -3.912585e-12
[191] -2.528435e-16 -5.324530e-13 -2.777717e-16 -3.917817e-08 -6.658627e-09
[196] -2.497941e-10 -4.336725e-15 -2.435416e-11 -2.557489e-09 -3.032714e-06
[201]  1.534846e-05 -1.247236e-14 -1.553229e-17 -3.853938e-08 -1.009625e-18
[206] -2.604916e-05 -5.072074e-13 -3.359444e-12 -2.449666e-07 -1.412503e-16
[211] -2.856976e-05 -1.045999e-09 -4.681019e-11 -8.254686e-13 -7.341053e-14
[216] -3.655204e-15 -9.492734e-11 -1.310170e-11 -5.650526e-15 -5.399780e-11
[221] -1.842718e-17 -2.511393e-10 -2.302084e-18 -5.898977e-07 -6.188258e-11
[226] -6.812241e-06 -3.319868e-09 -6.496955e-16 -1.438220e-08 -3.515950e-12
[231]  1.256874e-05 -1.736526e-16 -2.462690e-06 -1.097390e-09 -8.846154e-15
[236] -2.455956e-14 -3.972869e-08 -2.781746e-05 -1.496759e-16 -1.164443e-17
[241] -6.219744e-09 -1.957925e-16 -2.893186e-05 -5.919852e-18 -1.700308e-07
[246] -1.097467e-10 -4.447926e-07 -4.892119e-12 -1.668324e-18 -3.771759e-14
[251]  4.595299e-05 -2.426567e-11 -9.201865e-14 -3.578687e-13 -2.285925e-12
[256] -1.544877e-06 -2.995456e-05 -5.734980e-11 -2.888631e-18 -2.810824e-12
[261] -1.835560e-05 -1.150761e-16 -9.503163e-07 -3.446850e-17 -3.532268e-17
[266] -1.336760e-17 -3.732196e-07 -5.007189e-08 -3.870618e-15 -4.442837e-14
[271] -4.677521e-07
```

#==

Exercise 6:

#==

#1. Propose a model specification. Write the Likelihood function.

```
datstu_quality22=cbind(datstu_20000[,1],datstu_20000[,3])
datstu_quality22 = as.data.frame(datstu_quality22)
class(datstu_quality22)
like_fun2 = function(param,dat_stu)
{
  quality =  as.vector(dat_stu$quality)
  cats = as.vector(sort(unique(dat_stu[,1])))
  ch = as.vector(match(dat_stu[,1], cats))
  ni = nrow(dat_stu)
  nj = length(cats)
  ut = mat.or.vec(ni,nj)
  # multinomial logit
  pn1    = param[1:nj]
  pn2    = param[(nj+1):(2*nj)]
```

```
  for (j in 1:nj)
  {
    ut[,j] = quality*pn1[j] + pn2[j]
  }
  prob   = exp(ut)             # exp(XB)
  #sprob  = rowsums(prob)      # sum_j exp(XB) denominator
  prob   = sweep(prob,MARGIN=1,FUN="/",STATS=rowSums(prob))
  # match prob to actual choices
  probc = NULL
  for (i in 1:ni)
  {
    probc[i] = prob[i,ch[i]]
  }
  probc[probc>0.999999] = 0.999999
  probc[probc<0.000001] = 0.000001
  like = sum(log(probc)) / ni
  return(-like)
}
nj = length(as.vector(unique(datstu_quality22[,1])))
print(nj)
```

```
> print(nj)
[1] 271
```

```
res7 = optim(runif(2 * nj,min=-0.1,max=0),
             fn=like_fun2,method="BFGS",control=list(trace=5,REPORT=1,maxit=10),
             dat_stu=datstu_quality22,hessian=FALSE)
```

#2.Estimate parameters and compute marginal effect of the proposed model.

The parameters for the 271 choices are as follows. Since each choice has two parameters, there are 542 values in total.

```
> parameters2=res7$par
>
> parameters2
  [1] -0.0805169798 -0.0290842644 -0.0835719036 -0.0124988048 -0.0038820924 -0.0815625860 -0.0027585512
  [8] -0.0085663173 -0.0370953968 -0.0767890580 -0.0120627413 -0.0163025727 -0.0177888839 -0.0376932678
 [15] -0.0749930287 -0.0109735174 -0.0192252761 -0.0235505933 -0.0579799464 -0.0466196967 -0.0677214689
 [22] -0.0778943607 -0.0265868243 -0.0973306209 -0.0081546675 -0.0074337660 -0.0468880308 -0.0154937576
 [29] -0.0158006664 -0.0346670103 -0.0500203461 -0.0047580353 -0.0586749595 -0.0717161126 -0.0115252199
 [36] -0.0075075048 -0.0100370096 -0.0291694537 -0.0190648141 -0.0885112881 -0.0224155589 -0.0546268431
 [43] -0.0579966993 -0.0913748672 -0.0936896293 -0.0214348887 -0.0588256577 -0.0503124317 -0.0652577544
 [50] -0.0790153952 -0.0575942758 -0.0492667606 -0.0584839401 -0.0191327827 -0.0675341109 -0.0702754534
 [57] -0.0628675201 -0.0944030414 -0.0508350027 -0.0807898537 -0.0152984906 -0.0161178146 -0.0835679211
 [64] -0.0900124387 -0.0464756488 -0.0085505935 -0.0174036121 -0.0576320126 -0.0129103769 -0.0461618436
 [71] -0.0284939161 -0.0368886701 -0.0540588008 -0.0968115492 -0.0815876003 -0.0324609118 -0.0578081684
 [78] -0.0090664498 -0.0399517797 -0.0730825252 -0.0589169113 -0.0240576015 -0.0115486946 -0.0126587862
 [85] -0.0798810635 -0.0368360165 -0.0959798803 -0.0059569130 -0.0750802246 -0.0112883945 -0.0587500371
 [92] -0.0907246657 -0.0231575039 -0.0966965820 -0.0616415073 -0.0202521467 -0.0243914325 -0.0377806787
 [99] -0.0246776900 -0.0162674288 -0.0469039473 -0.0791616862 -0.0550078302 -0.0100124102 -0.0841623954
[106] -0.0892059748 -0.0877307604 -0.0866255051 -0.0579624237 -0.0984239819 -0.0781893207 -0.0487405331
[113] -0.0459577808 -0.0324888760 -0.0941028214 -0.0206656414 -0.0840626198 -0.0418119895 -0.0416003359
[120] -0.0116277748 -0.0832956638 -0.0759915248 -0.0453805083 -0.0589729527 -0.0636572760 -0.0219142331
[127] -0.0166237316 -0.0915032319 -0.0179115107 -0.0170170449 -0.0284849580 -0.0651207029 -0.0618807347
[134] -0.0402706325 -0.0274295885 -0.0176841819 -0.0776163213 -0.0655699600 -0.0610633750 -0.0258491551

[141] -0.0362479855 -0.0343809353 -0.0803110183 -0.0075073267 -0.0594323036 -0.0024736628 -0.0041770651
[148] -0.0771534950 -0.0553131636 -0.0548845267 -0.0527540776 -0.0594364553 -0.0214995092 -0.0847235790
[155] -0.0988851090 -0.0760310289 -0.0594916167 -0.0099982668 -0.0536173650 -0.0352728854 -0.0989545946
[162] -0.0235581214 -0.0301621293 -0.0422609319 -0.0441622343 -0.0340818048 -0.0168132854 -0.0249832480
[169] -0.0935142284 -0.0472024691 -0.0582869176 -0.0186309138 -0.0764600135 -0.0200480348 -0.0827788859
[176] -0.0886410741 -0.0606310015 -0.0549264439 -0.0792357138 -0.0265406902 -0.0979081168 -0.0165107186
[183] -0.0968371301 -0.0223995155 -0.0397878671 -0.0282473026 -0.0619445680 -0.0196367546 -0.0778127675
[190] -0.0219998949 -0.0472460516 -0.0205582256 -0.0812197590 -0.0602592938 -0.0507400046 -0.0187688079
[197] -0.0815459283 -0.0478554736 -0.0556738511 -0.0592060108 -0.0222076471 -0.0406348221 -0.0205093932
[204] -0.0681012794 -0.0635374328 -0.0215887280 -0.0184422636 -0.0494999321 -0.0326760921 -0.0960224668
[211] -0.0467096074 -0.0329629845 -0.0176264999 -0.0737944481 -0.0890899098 -0.0757869656 -0.0122982654
[218] -0.0481822915 -0.0134627386 -0.0515033138 -0.0113935382 -0.0060124603 -0.0379427402 -0.0736007182
[225] -0.0180873760 -0.0166633059 -0.0307207970 -0.0707980666 -0.0884465922 -0.0401941913 -0.0807020393
[232] -0.0930358899 -0.0698723819 -0.0693622397 -0.0505992465 -0.0369221582 -0.0445731213 -0.0390916756
[239] -0.0863644089 -0.0820773090 -0.0332725127 -0.0441753657 -0.0220389328 -0.0140835463 -0.0346017977
[246] -0.0391208519 -0.0906203235 -0.0169268300 -0.0357730888 -0.0142581711 -0.0208744750 -0.0503438867
[253] -0.0527170039 -0.0941117512 -0.0391032163 -0.0119408076 -0.0812313858 -0.0220399110 -0.0521405090
[260] -0.0184983598 -0.0293469530 -0.0939778509 -0.0814797524 -0.0573500369 -0.0848203410 -0.0927508229
[267] -0.0208064503 -0.0201623344 -0.0795891713 -0.0597832817 -0.0836430232 -0.0996289591 -0.0287641205
[274] -0.0352902124 -0.0994828813 -0.0495083476 -0.0640817949 -0.0789782523 -0.0509963704 -0.0825110057
[281] -0.0795676059 -0.0411224385 -0.0140293937 -0.0322059190 -0.0807278515 -0.0380134898 -0.0016106642
[288] -0.0569476419 -0.0624897497 -0.0419606635 -0.0430375368 -0.0945483409 -0.0839968598 -0.0079381313
[295] -0.0841207062 -0.0693493820 -0.0366783645 -0.0663963049 -0.0572408325 -0.0790648105 -0.0759979987
[302] -0.0984890739 -0.0827285925 -0.0930861928 -0.0614886531 -0.0992430104 -0.0336866289 -0.0953495011
[309] -0.0891481185 -0.0723084853 -0.0371546021 -0.0247293155 -0.0991251521 -0.0117658142 -0.0468450753
[316] -0.0029098135 -0.0212442213 -0.0149956446 -0.0794261357 -0.0897569576 -0.0336479624 -0.0629101165
```

```
[323] -0.0787300029 -0.0956216792 -0.0061351573 -0.0507522210 -0.0152233793 -0.0138967650 -0.0556474551
[330] -0.0869880611 -0.0254632180 -0.0613974111 -0.0030244417 -0.0525185125 -0.0071381671 -0.0408607971
[337] -0.0739308655 -0.0594371403 -0.0776439882 -0.0067285365 -0.0688922098 -0.0852571630 -0.0764960970
[344] -0.0664594575 -0.0257372048 -0.0501305653 -0.0635572019 -0.0975664335 -0.0915120140 -0.0597780801
[351] -0.0551058198 -0.0942908313 -0.0346770672 -0.0647195672 -0.0367607180 -0.0684034778 -0.0232485057
[358] -0.0424925138 -0.0131059729 -0.0216503347 -0.0545883517 -0.0764641425 -0.0107016201 -0.0981486378
[365] -0.0576938733 -0.0851185421 -0.0888331541 -0.0467531139 -0.0883845937 -0.0802643945 -0.0597930319
[372] -0.0841730055 -0.0743105697 -0.0820903035 -0.0231696073 -0.0943664812 -0.0191466538 -0.0011398623
[379] -0.0511112063 -0.0885478424 -0.0652041717 -0.0626778953 -0.0889453803 -0.0385837803 -0.0406272215
[386] -0.0753892416 -0.0485295519 -0.0987940325 -0.0272217268 -0.0823819132 -0.0972093091 -0.0415445035
[393] -0.0138141702 -0.0049038437 -0.0111183104 -0.0933485341 -0.0219417008 -0.0621481415 -0.0346983770
[400] -0.0608349465 -0.0318196095 -0.0065871442 -0.0771011473 -0.0698973947 -0.0641118831 -0.0542752443
[407] -0.0772903248 -0.0048175151 -0.0208622931 -0.0733259976 -0.0301019549 -0.0071591354 -0.0882057480
[414] -0.0350329323 -0.0359508007 -0.0399321886 -0.0800027461 -0.0428507728 -0.0249793426 -0.0540907384
[421] -0.0597418532 -0.0236266131 -0.0653849249 -0.0627831725 -0.0216800976 -0.0781934784 -0.0585242543
[428] -0.0427407504 -0.0764212850 -0.0642726010 -0.0925551887 -0.0693415682 -0.0547219211 -0.0349805732
[435] -0.0842701963 -0.0838716211 -0.0805040293 -0.0464761682 -0.0652628213 -0.0719538583 -0.0924417763
[442] -0.0591092367 -0.0770143050 -0.0191217317 -0.0617828396 -0.0445313028 -0.0997391880 -0.0652449973
[449] -0.0132977031 -0.0420533161 -0.0724459112 -0.0161346111 -0.0975234615 -0.0435291680 -0.0064171491
[456] -0.0920676060 -0.0872572392 -0.0163670703 -0.0003327359 -0.0789028455 -0.0911108177 -0.0301732138
[463] -0.0237112853 -0.0226728038 -0.0320570702 -0.0552631304 -0.0829567226 -0.0996163799 -0.0074134500
[470] -0.0309732995 -0.0303138880 -0.0554380357 -0.0742703013 -0.0820272289 -0.0322957710 -0.0299953837
[477] -0.0377705127 -0.0867148610 -0.0678707953 -0.0571134627 -0.0141339871 -0.0737088609 -0.0023220231
[484] -0.0387689074 -0.0905593164 -0.0509685764 -0.0324901022 -0.0196254901 -0.0690328422 -0.0787721482
[491] -0.0559244266 -0.0418268251 -0.0348444596 -0.0297192157 -0.0713075705 -0.0895255097 -0.0061332945
[498] -0.0191609108 -0.0859731513 -0.0257291893 -0.0384963959 -0.0924106747 -0.0192276699 -0.0447870615
[505] -0.0577099419 -0.0933920448 -0.0243998343 -0.0870276537 -0.0508893376 -0.0968403347 -0.0539137580
[512] -0.0552843519 -0.0537643601 -0.0463033533 -0.0485995999 -0.0611790300 -0.0374855878 -0.0995248593
[519] -0.0190750693 -0.0805165317 -0.0620104768 -0.0488199215 -0.0073981321 -0.0368316867 -0.0604855914
[526] -0.0265324240 -0.0206583603 -0.0677717576 -0.0532574663 -0.0491267849 -0.0945974422 -0.0021462692
[533] -0.0074950708 -0.0324292160 -0.0952392595 -0.0820271885 -0.0648987932 -0.0274671872 -0.0699405142
[540] -0.0272568778 -0.0883822361 -0.0957945222
```

#marginal effect

```
#marginal effect
#after we get the parameters, write it in the matrix way
beta1 = mat.or.vec(2,nj)
beta1[1,] = res7$par[1:271]
beta1[2,] = res7$par[272:542]
# build a matrix with columns "1" and scores
X1=cbind(rep(1,20000),datstu_quality22$quality)
#creat a matrix with 20000 row and 271 columns
a=mat.or.vec(20000,271)
#the sum of the exponential function
exponential1=exp(X1%*%beta1[,2:271])
rowSums(exponential1)
sum1=as.matrix(rowSums(exponential1))
a1=mat.or.vec(20000,271)
#give value for the matrix a
a1[,1]=1/(1+sum1)
a1[,2]=exp(X1 %*% beta1[,2])/(1+sum1)

for (i in 1:270){
  value1=exp(X1 %*% beta1[,i+1])/(1+sum1)
  a1[,i+1]=value1
}
```

```
#creat a 2000*271 matrix with the first column 0
b1=mat.or.vec(20000,271)
bbeta1=c(res7$par[272:542])
for (i in 1:20000) {
  b1[i,]=a1[i,]*(bbeta1-sum(a1[i,]*bbeta1))
}

marginal_effect2=apply(b1, 2, mean)
marginal_effect2
```

```
> marginal_effect2=apply(b1, 2, mean)
> marginal_effect2
```

The marginal effects for the 271 choices are as follows.

```
> marginal_effect2
  [1] -7.484123e-03 -2.390361e-10 -7.865165e-07  4.614779e-06 -1.373705e-06 -2.519066e-11 -3.076397e-06
  [8] -5.917961e-13 -5.772958e-10 -5.204545e-10  1.908579e-04 -2.922553e-06 -1.054999e-06 -1.385346e-11
 [15] -1.645999e-08 -3.037614e-10  4.541009e-05 -4.943904e-09 -1.355938e-11 -5.847631e-12  1.117766e-04
 [22] -1.238590e-07 -1.335293e-09  1.382556e-04 -1.138857e-09 -1.810152e-08 -8.917686e-14 -2.997125e-07
 [29]  1.575716e-04 -1.261839e-07 -6.695243e-14 -3.305017e-13 -1.872015e-06 -5.675196e-13 -3.300542e-06
 [36]  8.746853e-05 -1.654017e-15 -8.608552e-14 -1.148276e-09 -3.478407e-12 -3.171142e-13 -1.466733e-11
 [43] -1.439296e-06 -4.553367e-11 -7.566523e-08 -7.147101e-09 -1.213053e-15 -9.444445e-09 -6.236063e-13
 [50]  2.056311e-03 -8.183272e-09 -5.877750e-14 -7.784369e-10 -2.231238e-09 -2.437671e-13 -4.082225e-11
 [57] -4.203316e-07 -2.622400e-06 -8.557110e-12 -3.651182e-10 -5.562895e-13 -2.942916e-11 -2.562928e-06
 [64] -2.238368e-15 -2.537219e-09 -5.153847e-07 -8.425093e-11 -8.666707e-13 -1.839969e-14 -1.550274e-13
 [71] -1.182586e-14 -7.332761e-13 -1.039333e-11 -7.916992e-07 -3.891715e-07 -7.851798e-08 -1.307466e-13
 [78] -1.742249e-15 -2.860293e-12  8.637603e-04 -3.389137e-12 -3.575937e-08 -3.391993e-08 -3.134878e-06
 [85] -3.991396e-15 -1.178977e-13 -5.991717e-12 -2.389366e-11 -1.268374e-14  1.267144e-06 -3.062341e-06
 [92] -5.629665e-12 -3.344935e-11 -1.719844e-14 -2.988411e-06 -3.733778e-10 -1.542228e-14 -4.367474e-15
 [99] -1.635295e-12 -1.139559e-10 -4.003609e-11 -5.031386e-09 -6.927629e-13 -3.568097e-08  1.797007e-04
[106] -6.150863e-11 -9.632555e-09 -1.454031e-15 -1.420611e-13 -6.985828e-08 -1.170160e-07 -1.183528e-08
[113] -1.843495e-12 -1.144691e-15 -2.531589e-07 -5.776331e-07 -4.354345e-11 -1.659894e-15 -1.182862e-08
[120] -6.457970e-09 -4.562861e-08  1.827523e-03 -6.345626e-10 -1.091552e-06 -9.357234e-12 -1.434644e-08
[127] -5.016388e-10 -4.740471e-10 -2.103123e-13  3.873003e-04 -2.998987e-06 -2.819441e-08 -6.022279e-10
[134] -8.784341e-08 -3.329480e-10 -5.063970e-11 -9.932578e-07 -8.919371e-12 -9.858864e-14 -6.083368e-15
[141] -2.799538e-15 -5.772552e-07 -2.030111e-07 -5.654021e-11 -3.688502e-08 -1.578981e-13 -1.023253e-12
[148] -6.937399e-10 -3.168280e-06  1.194316e-06  4.067462e-06 -3.363985e-13 -1.049329e-14 -7.596382e-09
[155] -1.133947e-07 -1.303580e-07 -1.448439e-08 -3.021797e-06 -5.755079e-09  8.236247e-07  3.374682e-04
[162] -2.087781e-07 -3.689133e-09  6.994271e-04 -4.834128e-10 -3.260318e-15 -2.150328e-08 -6.301036e-08
[169] -5.525051e-08 -6.341674e-12 -1.250661e-13 -1.205452e-11 -3.082910e-06 -1.145882e-07 -4.254075e-13
[176] -1.259013e-06  1.817159e-05 -5.976387e-15 -5.999968e-10 -1.941523e-15 -1.475170e-08 -1.416555e-15
[183] -1.012444e-13 -5.572150e-08 -4.477261e-15 -1.947149e-14 -6.939869e-13 -2.324780e-06 -2.928412e-09
[190] -2.619615e-13 -3.399910e-10 -2.890926e-12 -1.813283e-07 -9.074111e-08 -7.170894e-14 -5.384420e-11
[197] -8.177243e-09 -4.918380e-13 -5.612593e-09 -1.199673e-07 -2.081353e-07 -6.366442e-14 -5.119298e-15


[204] -7.397196e-14 -1.214181e-08 -7.660839e-09 -3.387058e-13 -2.223511e-11  5.617950e-05 -5.724496e-14
[211] -2.360224e-09 -1.183385e-06 -1.497575e-09 -2.986234e-12 -1.759033e-13 -1.194618e-08 -4.705822e-12
[218] -3.297171e-06 -2.771379e-13 -7.084034e-07 -3.139013e-06 -2.741107e-12 -3.275807e-06 -6.735622e-15
[225] -1.054108e-14 -1.019616e-14 -1.524278e-14 -2.451005e-09 -6.108697e-13 -2.062522e-08 -2.352981e-14
[232]  3.915930e-04 -2.372158e-10 -4.475942e-12 -1.342137e-14 -8.108458e-11 -3.499572e-15 -2.602992e-15
[239] -5.213536e-09 -4.933586e-08 -9.651178e-07 -4.018161e-12 -3.261914e-06 -4.889181e-07 -1.408699e-15
[246] -2.887691e-14 -1.976112e-11 -3.162485e-06 -8.049469e-11 -4.250446e-12 -7.023873e-08 -1.284767e-12
[253] -2.115107e-06 -2.382820e-14 -2.372720e-15 -8.613868e-12 -1.306165e-07 -9.598381e-13 -2.672067e-08
[260] -1.068568e-11 -5.870421e-09 -4.327408e-12 -2.291378e-08 -1.601902e-11 -1.788652e-08 -7.785780e-09
[267] -1.190191e-11 -5.538136e-15 -1.682533e-12 -2.221238e-13 -3.066801e-14
```

#==
Exercise 7:
#==
In this exercise, we are interested in the effect of excluding choices where the program is "Others".

#1.Explain and justify, which model (first or second model) you think is appropriate to conduct this exercise.

Answer: I think the second model is appropriate, because in this question, it deleted the choices of "others", the choices of each individual have not changed. We want to compare different effects of different choices. The type of choice has changed, so it is more suitable to estimate with the average score of each choice (quality).

Delete program with others. The new dataset is datstu7.

```
datstu7=cbind(datstu7[,1],datstu7[,3],datstu7[,24])
datstu7 = datstu[-which(datstu$program_new == "others"),]
datstu7 = as.data.frame(datstu7)
```

#the first model

```
datstu7_1=cbind(datstu7[,1],datstu7[,3])
datstu7_1 = as.data.frame(datstu7_1)
class(datstu7_1)
colnames(datstu7_1)=c("choice_rev","score")
```

	choice_rev	score
1	100 arts	236
2	100 arts	235
3	100 arts	217
4	100 arts	236
5	100 arts	259
6	100 arts	276
7	100 arts	222
8	100 arts	224
9	100 arts	248
10	100 arts	233
11	100 arts	241
12	100 arts	224
13	100 arts	227
14	100 arts	232

Showing 1 to 15 of 116,929 entries, 2 total columns

```r
like_fun = function(param,dat_stu)
{
  score =  as.vector(dat_stu$score)
  cats = as.vector(sort(unique(dat_stu[,1])))
  ch = as.vector(match(dat_stu[,1], cats))
  ni = nrow(dat_stu)
  nj = length(cats)
  ut = mat.or.vec(ni,nj)
  # multinomial logit
  pn1    = param[1:nj]
  pn2    = param[(nj+1):(2*nj)]
  for (j in 1:nj)
  {
    ut[,j] = score*pn1[j] + pn2[j]
  }
  prob   = exp(ut)            # exp(XB)
  #sprob  = rowsums(prob)      # sum_j exp(XB) denominator
  prob   = sweep(prob,MARGIN=1,FUN="/",STATS=rowSums(prob))
  # match prob to actual choices
  probc = NULL
  for (i in 1:ni)
  {
    probc[i] = prob[i,ch[i]]
  }
  probc[probc>0.999999] = 0.999999
  probc[probc<0.000001] = 0.000001
  like = sum(log(probc)) / ni
  return(-like)
}
```

```r
nj = length(as.vector(unique(datstu7_1[,1])))
print(nj)
res8 = optim(runif(2 * nj,min=-0.1,max=0),fn=like_fun,
             method="BFGS",control=list(trace=5,REPORT=1,maxit=10),
             dat_stu=datstu7_1,hessian=FALSE)
```

#the second model

```r
datstu7_2=cbind(datstu7[,1],datstu7[,24])
datstu7_2 = as.data.frame(datstu7_2)
class(datstu7_2)
colnames(datstu7_2)=c("choice_rev","quality")
```

```r
like_fun2 = function(param,dat_stu)
{
  quality =  as.vector(dat_stu$quality)
  cats = as.vector(sort(unique(dat_stu[,1])))
  ch = as.vector(match(dat_stu[,1], cats))
  ni = nrow(dat_stu)
  nj = length(cats)
  ut = mat.or.vec(ni,nj)
  # multinomial logit
  pn1    = param[1:nj]
  pn2    = param[(nj+1):(2*nj)]
  for (j in 1:nj)
  {
    ut[,j] = quality*pn1[j] + pn2[j]
  }
  prob   = exp(ut)            # exp(XB)
  #sprob  = rowsums(prob)      # sum_j exp(XB) denominator
  prob   = sweep(prob,MARGIN=1,FUN="/",STATS=rowSums(prob))
  # match prob to actual choices
  probc = NULL
  for (i in 1:ni)
  {
    probc[i] = prob[i,ch[i]]
  }
  probc[probc>0.999999] = 0.999999
  probc[probc<0.000001] = 0.000001
  like = sum(log(probc)) / ni
  return(-like)
}
```

```r
nj = length(as.vector(unique(datstu_quality22[,1])))
print(nj)

res9 = optim(runif(2 * nj,min=-0.1,max=0),
             fn=like_fun2,method="BFGS",control=list(trace=5,REPORT=1,maxit=10),
             dat_stu=datstu7_2,hessian=FALSE)
```