#ECON613 A3 Mengzhi Chen
#==========

```
#Set working directory
setwd("/Users/halcyonchan/Desktop/Econ613/A3/Data")
library(data.table)
library(tidyverse)
```

#Exercise1==========
#1.1==========

```
#Calculate the following statistics.
#Read the file
datstu = fread("datstu_v2.csv")
datjss = fread("datjss.csv")
datsss = fread("datsss.csv")

#Number of students, schools, programs
#Calculate the number of students
stu_num = length(datstu$V1)

> stu_num #340823
[1] 340823

#Combine (school,program) into a long matrix
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)
choice = rbind(choice1,choice2,choice3,choice4,choice5,choice6)
#Calculate the number of schools that students apply using dataset datstu
school_num1 = length(unique(choice[,1]))

> school_num1 #641
[1] 641

#Calculate the total number of schools using dataset datsss
school2 = unique(datsss[,3])
school_num2 = nrow(school2)

> school_num2 #898
[1] 898

#Calculate the number of programs
program_num = length(unique(choice[,2]))

> program_num #33
[1] 33
```

#1.2==========

```r
#Number of choices (school, program)
#Hint: Create a matrix of school, programs. Convert data from Wide to Long.
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)
choice = unique(rbind(choice1,choice2,choice3,choice4,choice5,choice6))
colnames(choice)=c('schoolcode','choicepgm')
choice_num = nrow(choice)
```

```
> choice_num #3086
[1] 3086
```

#1.3==========

```r
#Number of students applying to at least one senior high schools in the same district to home
#(Suppose students live in the same district to their junior high schools)
#Choose the longest schoolname in dataset datsss
datsss = datsss %>%
  group_by(schoolcode) %>%
  filter(nchar(schoolname) == max(nchar(schoolname)))
datsss1 = unique(datsss[,c('schoolcode','sssdistrict')])

#match each schoolcode to its corresponding district
colnames(datsss1)=c('schoolcode1','sssdistrict1')
merge1 = merge(datstu,datsss1,by = c('schoolcode1'),allow.cartesian=TRUE)
colnames(datsss1)=c('schoolcode2','sssdistrict2')
merge2 = merge(merge1,datsss1,by = c('schoolcode2'),allow.cartesian=TRUE)
colnames(datsss1)=c('schoolcode3','sssdistrict3')
merge3 = merge(merge2,datsss1,by = c('schoolcode3'),allow.cartesian=TRUE)
colnames(datsss1)=c('schoolcode4','sssdistrict4')
merge4 = merge(merge3,datsss1,by = c('schoolcode4'),allow.cartesian=TRUE)
colnames(datsss1)=c('schoolcode5','sssdistrict5')
merge5 = merge(merge4,datsss1,by = c('schoolcode5'),allow.cartesian=TRUE)
colnames(datsss1)=c('schoolcode6','sssdistrict6')
merge6 = merge(merge5,datsss1,by = c('schoolcode6'),allow.cartesian=TRUE)

#If the student apply to at least one senior high schools in the same district to home, set "same_dist" as 1
#If there is no senior high schools in the same district to home, set "same_dist" as 0
merge6[which(merge6[,'jssdistrict'] == merge6[,'sssdistrict1'] | merge6[,'jssdistrict'] == merge6[,'sssdistrict2'] |
               merge6[,'jssdistrict'] == merge6[,'sssdistrict3'] | merge6[,'jssdistrict'] == merge6[,'sssdistrict4'] |
               merge6[,'jssdistrict'] == merge6[,'sssdistrict5'] | merge6[,'jssdistrict'] == merge6[,'sssdistrict6']),'same_dist'] = 1
#Sum the column of "same_dist"
sum(merge6[,'same_dist'],na.rm=TRUE) #250826
```

```
> sum(merge6[,'same_dist'],na.rm=TRUE)
[1] 250826
```

#1.4==========

```r
#Number of students each senior high school admitted
#First, find the schools that each student admitted to
datstu[which(datstu$rankplace == "1"),'admitted'] = datstu[which(datstu$rankplace == "1"),5]
datstu[which(datstu$rankplace == "2"),'admitted'] = datstu[which(datstu$rankplace == "2"),6]
datstu[which(datstu$rankplace == "3"),'admitted'] = datstu[which(datstu$rankplace == "3"),7]
datstu[which(datstu$rankplace == "4"),'admitted'] = datstu[which(datstu$rankplace == "4"),8]
datstu[which(datstu$rankplace == "5"),'admitted'] = datstu[which(datstu$rankplace == "5"),9]
datstu[which(datstu$rankplace == "6"),'admitted'] = datstu[which(datstu$rankplace == "6"),10]
datstu = datstu %>% drop_na(admitted)
#Add one column of 1's for further counting
count = matrix(rep(1,nrow(datstu)))
admitstu = cbind(datstu,count)
#Count the number of students admitted of each school use function aggregate and by
admitstu_count = aggregate(x=admitstu[,20],by=list(admitstu$admitted),FUN=sum)
#Rename the columns
colnames(admitstu_count)=c('schoolcode','num_stu')
view(admitstu_count)
```

	schoolcode	num_stu
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

Showing 1 to 15 of 517 entries, 2 total columns

#1.5==========

```r
#The cutoff of senior high schools (the lowest score to be admitted)
admitstu_cutoff = aggregate(x=admitstu[,2],by=list(admitstu$admitted),FUN=min)
colnames(admitstu_cutoff)=c('schoolcode','cutoff')
view(admitstu_cutoff)
```

	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
11	10111	371
12	10112	316
13	10114	319
14	10115	274
15	10116	205

Showing 1 to 15 of 517 entries, 2 total columns

#1.6==========

```
#The quality of senior high schools (the average score of students admitted)
admitstu_quality = aggregate(x=admitstu[,2],by=list(admitstu$admitted),FUN=mean)
colnames(admitstu_quality)=c('schoolcode','quality')
view(admitstu_quality)
```

	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
11	10111	412.5100
12	10112	375.6133
13	10114	346.2229
14	10115	316.3361
15	10116	289.9574

Showing 1 to 15 of 517 entries, 2 total columns

#Exercise2==========

```r
#Create a school level dataset, where each row corresponds to a (school,program) with the following variables:
#1.the district where the school is located
#2.the latitude and longtitude 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)

#drop na in datsss and merge it with choice(school,program)
datsss1 = datsss[,c('schoolcode','sssdistrict','ssslong','ssslat')]
datsss1 = datsss1 %>% drop_na(ssslong)
datschool = merge(choice,datsss1,by = c('schoolcode'),all.x=TRUE)

#Add a column "admitpgm" (admitted program) in dataset datstu
datstu[which(datstu$rankplace == "1"),'admitpgm'] = datstu[which(datstu$rankplace == "1"),11]
datstu[which(datstu$rankplace == "2"),'admitpgm'] = datstu[which(datstu$rankplace == "2"),12]
datstu[which(datstu$rankplace == "3"),'admitpgm'] = datstu[which(datstu$rankplace == "3"),13]
datstu[which(datstu$rankplace == "4"),'admitpgm'] = datstu[which(datstu$rankplace == "4"),14]
datstu[which(datstu$rankplace == "5"),'admitpgm'] = datstu[which(datstu$rankplace == "5"),15]
datstu[which(datstu$rankplace == "6"),'admitpgm'] = datstu[which(datstu$rankplace == "6"),16]
#Add one column of 1's for further counting
count = matrix(rep(1,nrow(datstu)))
admitstu = cbind(datstu,count)
#Compute the cutoff of each choice
datschool_cutoff = aggregate(x=admitstu[,2],by=list(admitstu$admitted,admitstu$admitpgm),FUN=min)
colnames(datschool_cutoff) = c('schoolcode','choicepgm','score')
#Compute the quality of each choice
datschool_quality = aggregate(x=admitstu[,2],by=list(admitstu$admitted,admitstu$admitpgm),FUN=mean)
colnames(datschool_quality) = c('schoolcode','choicepgm','score')
#Compute the size of each choice
datschool_num = aggregate(x=admitstu[,21],by=list(admitstu$admitted,admitstu$admitpgm),FUN=sum)
colnames(datschool_num) = c('schoolcode','choicepgm','size')
```

```r
#merge these columns
datschool = merge(datschool,datschool_cutoff,by = c('schoolcode','choicepgm'),all.x=TRUE)
datschool = merge(datschool,datschool_quality,by = c('schoolcode','choicepgm'),all.x=TRUE)
datschool = merge(datschool,datschool_num,by = c('schoolcode','choicepgm'),all.x=TRUE)
#Drop the repeated rows using unique function
datschool = unique(datschool)
#Rename the columns
colnames(datschool) = c('schoolcode','choicepgm','district','longtitude','latitude','cutoff','quality','size')
```

Filter

	schoolcode	choicepgm	district	longtitude	latitude	cutoff	quality	size
1	100101		Wa Municipal	-2.2850304	10.030622	NA	NA	NA
4	100101	General Arts	Wa Municipal	-2.2850304	10.030622	198	244.3924	79
7	100101	Home Economics	Wa Municipal	-2.2850304	10.030622	199	229.4500	40
10	100101	Technical	Wa Municipal	-2.2850304	10.030622	201	235.1020	49
13	100102	Agriculture	Wa Municipal	-2.2850304	10.030622	273	292.5556	90
19	100102	Business	Wa Municipal	-2.2850304	10.030622	283	303.3444	90
25	100102	General Arts	Wa Municipal	-2.2850304	10.030622	291	311.1111	90
31	100102	General Science	Wa Municipal	-2.2850304	10.030622	273	298.4333	90
37	100102	Home Economics	Wa Municipal	-2.2850304	10.030622	262	278.8667	45
43	100102	Visual Arts	Wa Municipal	-2.2850304	10.030622	250	275.2000	45
49	100103	Agriculture	Wa Municipal	-2.2850304	10.030622	NA	NA	NA
50	100103	Business	Wa Municipal	-2.2850304	10.030622	NA	NA	NA
51	100103	General Arts	Wa Municipal	-2.2850304	10.030622	NA	NA	NA
52	100104	Business	Wa Municipal	-2.2850304	10.030622	NA	NA	NA
55	100104	General Arts	Wa Municipal	-2.2850304	10.030622	319	337.4444	45
58	100104	General Science	Wa Municipal	-2.2850304	10.030622	313	334.0000	45
61	100104	Home Economics	Wa Municipal	-2.2850304	10.030622	282	309.3556	45
64	100105		Wa Municipal	-2.2850304	10.030622	NA	NA	NA
67	100105	Business	Wa Municipal	-2.2850304	10.030622	251	268.0125	80
70	100105	General Arts	Wa Municipal	-2.2850304	10.030622	258	274.7375	80

Showing 1 to 20 of 3,086 entries, 8 total columns

#Exercise3==========

#3.1==========

```
#Using the formula
#dist(sss, jss) = sqrt(69.172 * (ssslong - jsslong) * cos(jsslat/57.3))^2 + (69.172 * (sssslat - jsslat))^2)
#where ssslong and sssslat 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.

datstu = fread("datstu_v2.csv")
datjss1 = datjss[,c(2,3,4)]
datsss1 = datsss[,c(3,4,5,6)]
#For schoolcode1 for each student
choice_jss1 = cbind(datstu$V1,datstu$schoolcode1,datstu$choicepgm1,datstu$jssdistrict)
colnames(choice_jss1) = c('student','schoolcode','choicepgm','jssdistrict')
#merge the choices with jsslong and jsslat according to 'jssdistrict', as well as sssdistrict and their longtitude and latitude
choice_dist1 = merge(choice_jss1,datjss1,by = c('jssdistrict'),all.x=TRUE)
choice_dist1 = unique(merge(choice_dist1,datsss1,by = c('schoolcode'),all.x=TRUE))
#Calculate dist using the formula
choice_dist1[,'dist'] = sqrt((69.172*(choice_dist1[,'ssslong']-choice_dist1[,'point_x'])
                              *cos(choice_dist1[,'point_y']/57.3))^2+(69.172*(choice_dist1[,'ssslat']-choice_dist1[,'point_y']))^2)
view(choice_dist1)
```

	schoolcode	jssdistrict	student	choicepgm	point_x	point_y	sssdistrict	ssslong	ssslat	dist
1	100101	Lawra	41963	Technical	-2.8009412	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
4	100101	Wa West (Wechiau)	44015	Home Economics	-2.6441252	9.915487	Wa Municipal	-2.28503	10.03062	25.73180
7	100101	Sunyani	116011	Home Economics	-2.3274233	7.344678	Wa Municipal	-2.28503	10.03062	185.81485
10	100101	Wa East (Funsi)	162673	Home Economics	-1.8335643	9.952777	Wa Municipal	-2.28503	10.03062	31.22667
13	100101	Tamale	130842	Technical	-0.7843482	9.383351	Wa Municipal	-2.28503	10.03062	111.77542
16	100101	Accra Metropolitan	334669	General Arts	-0.1971153	5.607396	Wa Municipal	-2.28503	10.03062	338.04309
19	100101	Wa West (Wechiau)	51868	Home Economics	-2.6441252	9.915487	Wa Municipal	-2.28503	10.03062	25.73180
22	100101	Dormaa (Dormaa Ahenkro)	46913	General Arts	-2.7538404	7.222849	Wa Municipal	-2.28503	10.03062	196.86570
25	100101	Kintampo South	334699	General Arts	-1.7633157	7.901240	Wa Municipal	-2.28503	10.03062	151.56893
28	100101	Wa Municipal	8442	General Arts	-2.2850304	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
31	100101	Wa Municipal	89605	Home Economics	-2.2850304	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
34	100101	Ejura/Sekyedumase (Ejura)	169658	General Arts	-1.3679653	7.462874	Wa Municipal	-2.28503	10.03062	188.42420
37	100101	Nadowli	89415	General Arts	-2.4276245	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
40	100101	Wa East (Funsi)	73512	Home Economics	-1.8335643	9.952777	Wa Municipal	-2.28503	10.03062	31.22667
43	100101	Jirapa/Lambusie (Jirapa)	170304	Home Economics	-2.5833225	10.692035	Wa Municipal	-2.28503	10.03062	50.04265
46	100101	Jirapa/Lambusie (Jirapa)	8237	General Arts	-2.5833225	10.692035	Wa Municipal	-2.28503	10.03062	50.04265
49	100101	West Gonja (Damongo)	2193	Technical	-1.7004058	9.505651	Wa Municipal	-2.28503	10.03062	53.93905
52	100101	Techiman	334607	Technical	-1.9448420	7.677615	Wa Municipal	-2.28503	10.03062	164.42439
55	100101	Techiman	334563	General Arts	-1.9448420	7.677615	Wa Municipal	-2.28503	10.03062	164.42439
58	100101	Jirapa/Lambusie (Jirapa)	858	Technical	-2.5833225	10.692035	Wa Municipal	-2.28503	10.03062	50.04265

Showing 1 to 20 of 340,823 entries, 10 total columns

```
#For schoolcode2 for each student
choice_jss2 = cbind(datstu$V1,datstu$schoolcode2,datstu$choicepgm2,datstu$jssdistrict)
colnames(choice_jss2) = c('student','schoolcode','choicepgm','jssdistrict')
choice_dist2 = merge(choice_jss2,datjss1,by = c('jssdistrict'),all.x=TRUE)
choice_dist2 = unique(merge(choice_dist2,datsss1,by = c('schoolcode'),all.x=TRUE))
choice_dist2[,'dist'] = sqrt((69.172*(choice_dist2[,'ssslong']-choice_dist2[,'point_x'])
                              *cos(choice_dist2[,'point_y']/57.3))^2+(69.172*(choice_dist2[,'ssslat']-choice_dist2[,'point_y']))^2)
view(choice_dist2)
```

	schoolcode	jssdistrict	student	choicepgm	point_x	point_y	sssdistrict	ssslong	ssslat	dist
1	100101	Lawra	47900	General Arts	-2.80094123	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
4	100101	Sawla-Tuna-Kalba	161158	Home Economics	-2.36116719	9.407022	Wa Municipal	-2.28503	10.03062	43.44745
7	100101	Asante Akim North (Kono…	40214	General Arts	-1.01796305	6.834004	Wa Municipal	-2.28503	10.03062	237.62464
10	100101	Wa West (Wechiau)	839	Home Economics	-2.64412522	9.915487	Wa Municipal	-2.28503	10.03062	25.73180
13	100101	Wa Municipal	123644	General Arts	-2.28503036	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
16	100101	Lawra	33960	Technical	-2.80094123	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
19	100101	East Dagomba (Yendi)	79802	Technical	-0.20084506	9.352131	Wa Municipal	-2.28503	10.03062	149.79353
22	100101	Wa Municipal	150871	Home Economics	-2.28503036	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
25	100101	Lawra	179649	Technical	-2.80094123	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
28	100101	Lawra	145072	Technical	-2.80094123	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
31	100101	Accra Metropolitan	141497	General Arts	-0.19711526	5.607396	Wa Municipal	-2.28503	10.03062	338.04309
34	100101	Wa Municipal	108746	Home Economics	-2.28503036	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
37	100101	Wa Municipal	114435	Technical	-2.28503036	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
40	100101	Wa East (Funsi)	65139	General Arts	-1.83356428	9.952777	Wa Municipal	-2.28503	10.03062	31.22667
43	100101	Nadowli	150445	Technical	-2.42762446	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
46	100101	Lawra	121188	Technical	-2.80094123	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
49	100101	Wa Municipal	33140	Technical	-2.28503036	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
52	100101	Nadowli	246090	General Arts	-2.42762446	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
55	100101	Tano North	334481	General Arts	-2.17906523	7.177094	Wa Municipal	-2.28503	10.03062	197.51817
58	100101	Wa West (Wechiau)	69247	Technical	-2.64412522	9.915487	Wa Municipal	-2.28503	10.03062	25.73180

Showing 1 to 20 of 340,823 entries, 10 total columns

```
#For schoolcode3 for each student
choice_jss3 = cbind(datstu$V1,datstu$schoolcode3,datstu$choicepgm3,datstu$jssdistrict)
colnames(choice_jss3) = c('student','schoolcode','choicepgm','jssdistrict')
choice_dist3 = merge(choice_jss3,datjss1,by = c('jssdistrict'),all.x=TRUE)
choice_dist3 = unique(merge(choice_dist3,datsss1,by = c('schoolcode'),all.x=TRUE))
choice_dist3['dist'] = sqrt((69.172*(choice_dist3['ssslong']-choice_dist3['point_x'])
                             *cos(choice_dist3['point_y']/57.3))^2+(69.172*(choice_dist3['ssslat']-choice_dist3['point_y']))^2)
view(choice_dist3)
```

	schoolcode	jssdistrict	student	choicepgm	point_x	point_y	sssdistrict	ssslong	ssslat	dist
1	100101	Wenchi	335489	General Arts	-1.9448420	7.677615	Wa Municipal	-2.28503	10.03062	164.42439
4	100101	Nadowli	106951	Home Economics	-2.4276245	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
7	100101	Wa Municipal	334117	Technical	-2.2850304	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
10	100101	Lawra	71413	Technical	-2.8009412	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
13	100101	Wa Municipal	49996	Technical	-2.2850304	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
16	100101	Kintampo	8622	General Arts	-1.5444263	8.371119	Wa Municipal	-2.28503	10.03062	125.48230
19	100101	Wa West (Wechiau)	70108	Home Economics	-2.6441252	9.915487	Wa Municipal	-2.28503	10.03062	25.73180
22	100101	Kumasi Metro	241809	General Arts	-1.5971872	6.682060	Wa Municipal	-2.28503	10.03062	236.39817
25	100101	Kwahu West (Nkawkaw)	334547	Technical	-0.8019751	6.527021	Wa Municipal	-2.28503	10.03062	262.91052
28	100101	Wa East (Funsi)	9624	Home Economics	-1.8335643	9.952777	Wa Municipal	-2.28503	10.03062	31.22667
31	100101	Nadowli	167013	Technical	-2.4276245	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
34	100101	Wa Municipal	50564	Technical	-2.2850304	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
37	100101	Dormaa (Dormaa Ahenkro)	82097	Technical	-2.7538404	7.222849	Wa Municipal	-2.28503	10.03062	196.86570
40	100101	Wa East (Funsi)	334522	General Arts	-1.8335643	9.952777	Wa Municipal	-2.28503	10.03062	31.22667
43	100101	Wa Municipal	247314	General Arts	-2.2850304	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
46	100101	Wa Municipal	76488	Technical	-2.2850304	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
49	100101	Sisala/Tumu (Tumu)	106683	General Arts	-1.7310659	10.574178	Wa Municipal	-2.28503	10.03062	53.22188
52	100101	Wa Municipal	334627	Home Economics	-2.2850304	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
55	100101	Jirapa/Lambusie (Jirapa)	334690	General Arts	-2.5833225	10.692035	Wa Municipal	-2.28503	10.03062	50.04265
58	100101	Sisala/Tumu (Tumu)	43726	Home Economics	-1.7310659	10.574178	Wa Municipal	-2.28503	10.03062	53.22188

Showing 1 to 20 of 340,823 entries, 10 total columns

```
#For schoolcode4 for each student
choice_jss4 = cbind(datstu$V1,datstu$schoolcode4,datstu$choicepgm4,datstu$jssdistrict)
colnames(choice_jss4) = c('student','schoolcode','choicepgm','jssdistrict')
choice_dist4 = merge(choice_jss4,datjss1,by = c('jssdistrict'),all.x=TRUE)
choice_dist4 = unique(merge(choice_dist4,datsss1,by = c('schoolcode'),all.x=TRUE))
choice_dist4['dist'] = sqrt((69.172*(choice_dist4['ssslong']-choice_dist4['point_x'])
                             *cos(choice_dist4['point_y']/57.3))^2+(69.172*(choice_dist4['ssslat']-choice_dist4['point_y']))^2)
view(choice_dist4)
```

	schoolcode	jssdistrict	student	choicepgm	point_x	point_y	sssdistrict	ssslong	ssslat	dist
1	100101	Lawra	10863	Technical	-2.80094123	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
4	100101	Lawra	334636	Technical	-2.80094123	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
7	100101	Wa West (Wechiau)	334593	General Arts	-2.64412522	9.915487	Wa Municipal	-2.28503	10.03062	25.73180
10	100101	Nadowli	47232	Home Economics	-2.42762446	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
13	100101	Wa East (Funsi)	153659	Technical	-1.83356428	9.952777	Wa Municipal	-2.28503	10.03062	31.22667
16	100101	Wa West (Wechiau)	40743	General Arts	-2.64412522	9.915487	Wa Municipal	-2.28503	10.03062	25.73180
19	100101	Wa Municipal	334400	General Arts	-2.28503036	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
22	100101	Lawra	18884	Technical	-2.80094123	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
25	100101	Techiman	300904	General Arts	-1.94484198	7.677615	Wa Municipal	-2.28503	10.03062	164.42439
28	100101	Wa Municipal	11331	General Arts	-2.28503036	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
31	100101	Wa Municipal	42091	Home Economics	-2.28503036	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
34	100101	Nadowli	334666	General Arts	-2.42762446	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
37	100101	Wa Municipal	122737	Home Economics	-2.28503036	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
40	100101	Tain	334379	General Arts	-2.28683090	7.948905	Wa Municipal	-2.28503	10.03062	143.99652
43	100101	Bosomtwe/Atwima/Kwan...	257563	Technical	-1.56275165	6.559323	Wa Municipal	-2.28503	10.03062	245.19298
46	100101	Wa East (Funsi)	72747	Home Economics	-1.83356428	9.952777	Wa Municipal	-2.28503	10.03062	31.22667
49	100101	Nadowli	300934	General Arts	-2.42762446	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
52	100101	Nadowli	334614	Technical	-2.42762446	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
55	100101	Wa Municipal	174997	Technical	-2.28503036	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
58	100101	Wa West (Wechiau)	246320	General Arts	-2.64412522	9.915487	Wa Municipal	-2.28503	10.03062	25.73180

Showing 1 to 20 of 340,823 entries, 10 total columns

```
#For schoolcode5 for each student
choice_jss5 = cbind(datstu$V1,datstu$schoolcode5,datstu$choicepgm5,datstu$jssdistrict)
colnames(choice_jss5) = c('student','schoolcode','choicepgm','jssdistrict')
choice_dist5 = merge(choice_jss5,datjss1,by = c('jssdistrict'),all.x=TRUE)
choice_dist5 = unique(merge(choice_dist5,datsss1,by = c('schoolcode'),all.x=TRUE))
choice_dist5['dist'] = sqrt((69.172*(choice_dist5['ssslong']-choice_dist5['point_x'])
                             *cos(choice_dist5['point_y']/57.3))^2+(69.172*(choice_dist5['ssslat']-choice_dist5['point_y']))^2)
view(choice_dist5)
```

	schoolcode	jssdistrict	student	choicepgm	point_x	point_y	sssdistrict	ssslong	ssslat	dist
1	100101	Nadowli	7043	Technical	-2.427624	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
4	100101	Wa East (Funsi)	15198	Home Economics	-1.833564	9.952777	Wa Municipal	-2.28503	10.03062	31.22667
7	100101	Sisala/Tumu (Tumu)	130239	Home Economics	-1.731066	10.574178	Wa Municipal	-2.28503	10.03062	53.22188
10	100101	Sisala/Tumu (Tumu)	73469	Home Economics	-1.731066	10.574178	Wa Municipal	-2.28503	10.03062	53.22188
13	100101	Lawra	128394	General Arts	-2.800941	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
16	100101	Wa West (Wechiau)	718	Home Economics	-2.644125	9.915487	Wa Municipal	-2.28503	10.03062	25.73180
19	100101	Lawra	54411		-2.800941	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
22	100101	Nadowli	225060	Technical	-2.427624	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
25	100101	Lawra	74963	General Arts	-2.800941	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
28	100101	Sisala west (Gwollu)	334668	Home Economics	-2.286119	10.757883	Wa Municipal	-2.28503	10.03062	50.30619
31	100101	Wa East (Funsi)	330681	General Arts	-1.833564	9.952777	Wa Municipal	-2.28503	10.03062	31.22667
34	100101	Nadowli	40420	Home Economics	-2.427624	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
37	100101	Wa East (Funsi)	321821	Home Economics	-1.833564	9.952777	Wa Municipal	-2.28503	10.03062	31.22667
40	100101	Nadowli	34380	Home Economics	-2.427624	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
43	100101	Lawra	120040	General Arts	-2.800941	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
46	100101	Wa Municipal	22242	Technical	-2.285030	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
49	100101	Nadowli	49560	Home Economics	-2.427624	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
52	100101	Wa East (Funsi)	64622	General Arts	-1.833564	9.952777	Wa Municipal	-2.28503	10.03062	31.22667
55	100101	Wa Municipal	334577	General Arts	-2.285030	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
58	100101	Wa West (Wechiau)	272658	Home Economics	-2.644125	9.915487	Wa Municipal	-2.28503	10.03062	25.73180

Showing 1 to 20 of 340,823 entries, 10 total columns

```
#For schoolcode6 for each student
choice_jss6 = cbind(datstu$V1,datstu$schoolcode6,datstu$choicepgm6,datstu$jssdistrict)
colnames(choice_jss6) = c('student','schoolcode','choicepgm','jssdistrict')
choice_dist6 = merge(choice_jss6,datjss1,by = c('jssdistrict'),all.x=TRUE)
choice_dist6 = unique(merge(choice_dist6,datsss1,by = c('schoolcode'),all.x=TRUE))
choice_dist6['dist'] = sqrt((69.172*(choice_dist6['ssslong']-choice_dist6['point_x'])
                             *cos(choice_dist6['point_y']/57.3))^2+(69.172*(choice_dist6['ssslat']-choice_dist6['point_y']))^2)
view(choice_dist6)
```

	schoolcode	jssdistrict	student	choicepgm	point_x	point_y	sssdistrict	ssslong	ssslat	dist
1	100101	Jirapa/Lambusie (Jirapa)	124338	Home Economics	-2.583323	10.692035	Wa Municipal	-2.28503	10.03062	50.04265
4	100101	Wa East (Funsi)	119958	Technical	-1.833564	9.952777	Wa Municipal	-2.28503	10.03062	31.22667
7	100101	Wa West (Wechiau)	91198	Home Economics	-2.644125	9.915487	Wa Municipal	-2.28503	10.03062	25.73180
10	100101	Lawra	146031	Technical	-2.800941	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
13	100101	Nadowli	114699	General Arts	-2.427624	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
16	100101	Sisala/Tumu (Tumu)	247096	General Arts	-1.731066	10.574178	Wa Municipal	-2.28503	10.03062	53.22188
19	100101	Wa West (Wechiau)	300965	Technical	-2.644125	9.915487	Wa Municipal	-2.28503	10.03062	25.73180
22	100101	Sisala/Tumu (Tumu)	100532	Home Economics	-1.731066	10.574178	Wa Municipal	-2.28503	10.03062	53.22188
25	100101	Wa Municipal	110286	Technical	-2.285030	10.030622	Wa Municipal	-2.28503	10.03062	0.00000
28	100101	Nadowli	110218	General Arts	-2.427624	10.362510	Wa Municipal	-2.28503	10.03062	24.92353
31	100101	Jirapa/Lambusie (Jirapa)	152297	Technical	-2.583323	10.692035	Wa Municipal	-2.28503	10.03062	50.04265
34	100101	Sisala west (Gwollu)	14158	General Arts	-2.286119	10.757883	Wa Municipal	-2.28503	10.03062	50.30619
37	100101	Sisala/Tumu (Tumu)	247065	General Arts	-1.731066	10.574178	Wa Municipal	-2.28503	10.03062	53.22188
40	100101	Wa West (Wechiau)	69247	Home Economics	-2.644125	9.915487	Wa Municipal	-2.28503	10.03062	25.73180
43	100101	Lawra	249916	General Arts	-2.800941	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
46	100101	Lawra	157524	Technical	-2.800941	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
49	100101	Wa West (Wechiau)	110266	Home Economics	-2.644125	9.915487	Wa Municipal	-2.28503	10.03062	25.73180
52	100101	Lawra	171081	General Arts	-2.800941	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
55	100101	Lawra	234448	Technical	-2.800941	10.546398	Wa Municipal	-2.28503	10.03062	50.03743
58	100101	Wa Municipal	321754	General Arts	-2.285030	10.030622	Wa Municipal	-2.28503	10.03062	0.00000

Showing 1 to 20 of 340,823 entries, 10 total columns

#Exercise4==========

#4.1==========

```
#Recode the schoolcode into its first three digits (substr). Call this new variable scode_rev.
scode_rev = substring(choice[,1],1,3)
choice_code = cbind(choice,scode_rev)
choice_code = data.frame(choice_code)
view(choice_code)
```

	schoolcode	choicepgm	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
16	80102	General Arts	801
17	10401	General Arts	104
18	60301	Agriculture	603
19	100102	General Arts	100
20	50501	Home Economics	505

Showing 1 to 20 of 3,086 entries, 3 total columns

#4.2==========

```
#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.
choice_code[which(choice_code[,2] == 'General Arts'),'pgm_rev'] = 'arts'
choice_code[which(choice_code[,2] == 'Visual Arts'),'pgm_rev'] = 'arts'
choice_code[which(choice_code[,2] == 'Business'),'pgm_rev'] = 'economics'
choice_code[which(choice_code[,2] == 'Home Economics'),'pgm_rev'] = 'economics'
choice_code[which(choice_code[,2] == 'General Science'),'pgm_rev'] = 'science'
choice_code[which(is.na(choice_code[,4])==TRUE),'pgm_rev'] = 'others'
view(choice_code)
```

	schoolcode	choicepgm	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
12	30603	Business	306	economics
13	80101	Business	801	economics
14	90301	Technical	903	others
15	40903	General Arts	409	arts
16	80102	General Arts	801	arts
17	10401	General Arts	104	arts
18	60301	Agriculture	603	others
19	100102	General Arts	100	arts
20	50501	Home Economics	505	economics

Showing 1 to 20 of 3,086 entries, 4 total columns

#4.3==========

```
choice_code[,'choice_rev'] = paste(choice_code[,'scode_rev'],choice_code[,'pgm_rev'])
view(choice_code)
```

	schoolcode	choicepgm	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
12	30603	Business	306	economics	306 economics
13	80101	Business	801	economics	801 economics
14	90301	Technical	903	others	903 others
15	40903	General Arts	409	arts	409 arts
16	80102	General Arts	801	arts	801 arts
17	10401	General Arts	104	arts	104 arts
18	60301	Agriculture	603	others	603 others
19	100102	General Arts	100	arts	100 arts
20	50501	Home Economics	505	economics	505 economics

Showing 1 to 20 of 3,086 entries, 5 total columns

#4.4==========

```
#Recalculate the cutoff and the quality for each recoded choice.
admitstu1 = admitstu[,c('score','admitted','admitpgm')]
admitstu1 = data.frame(admitstu1)
#Recode the schoolcode in dataset admitstu1
scode_rev = substring(admitstu1[,2],1,3)
admitstu1 = cbind(admitstu1,scode_rev)
#Recode the program in dataset admitstu1
admitstu1[which(admitstu1[,3] == 'General Arts'),'pgm_rev'] = 'arts'
admitstu1[which(admitstu1[,3] == 'Visual Arts'),'pgm_rev'] = 'arts'
admitstu1[which(admitstu1[,3] == 'Business'),'pgm_rev'] = 'economics'
admitstu1[which(admitstu1[,3] == 'Home Economics'),'pgm_rev'] = 'economics'
admitstu1[which(admitstu1[,3] == 'General Science'),'pgm_rev'] = 'science'
admitstu1[which(is.na(admitstu1[,5])==TRUE),'pgm_rev'] = 'others'
admitstu1[,'choice_rev'] = paste(admitstu1[,'scode_rev'],admitstu1[,'pgm_rev'])
admitstu2 = admitstu1[,c('score','choice_rev')]
#Use aggregate to calculate the cutoff and quality of each choice_rev
choice_rev_cutoff = aggregate(x=admitstu2[,'score'],by=list(admitstu2$choice_rev),FUN=min)
colnames(choice_rev_cutoff)=c('choice_rev','cutoff')
view(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

Showing 1 to 10 of 425 entries, 2 total columns

```
choice_rev_quality = aggregate(x=admitstu2['score'],by=list(admitstu2$choice_rev),FUN=mean)
colnames(choice_rev_quality)=c('choice_rev','quality')
view(choice_rev_quality)
```

	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

Showing 1 to 10 of 425 entries, 2 total columns

```
choice_rev = merge(choice_code,choice_rev_cutoff,by='choice_rev',all.x=TRUE)
choice_rev = merge(choice_rev,choice_rev_quality,by='choice_rev',all.x=TRUE)
view(choice_rev)
```

	choice_rev	schoolcode	choicepgm	scode_rev	pgm_rev	cutoff	quality
1	100 arts	100104	General Arts	100	arts	194	275.5233
2	100 arts	100301	General Arts	100	arts	194	275.5233
3	100 arts	100303	General Arts	100	arts	194	275.5233
4	100 arts	100401	Visual Arts	100	arts	194	275.5233
5	100 arts	100102	Visual Arts	100	arts	194	275.5233
6	100 arts	100302	General Arts	100	arts	194	275.5233
7	100 arts	100102	General Arts	100	arts	194	275.5233
8	100 arts	100105	General Arts	100	arts	194	275.5233
9	100 arts	100204	General Arts	100	arts	194	275.5233
10	100 arts	100503	General Arts	100	arts	194	275.5233
11	100 arts	100401	General Arts	100	arts	194	275.5233
12	100 arts	100201	General Arts	100	arts	194	275.5233
13	100 arts	100304	Visual Arts	100	arts	194	275.5233
14	100 arts	100502	General Arts	100	arts	194	275.5233
15	100 arts	100101	General Arts	100	arts	194	275.5233
16	100 arts	100103	General Arts	100	arts	194	275.5233
17	100 arts	100106	General Arts	100	arts	194	275.5233
18	100 arts	100202	General Arts	100	arts	194	275.5233
19	100 economics	100304	Home Economics	100	economics	195	264.4993
20	100 economics	100105	Business	100	economics	195	264.4993

Showing 1 to 20 of 3,086 entries, 7 total columns

#4.5==========

```r
#Consider the 20,000 highest score students.
#The rest of the assignment uses the recoded choices and the 20,000 highest score students.
admitstu = admitstu[order(admitstu$score,decreasing = TRUE),]
admitstu = admitstu[1:20000,]
#Process the data of the first choice of each student
admitstu1 = admitstu[,c('score','schoolcode1','choicepgm1')]
admitstu1 = data.frame(admitstu1)
#Recode the schoolcode in dataset admitstu1
scode_rev = substring(admitstu1[,2],1,3)
admitstu1 = cbind(admitstu1,scode_rev)
#Recode the program in dataset admitstu1
admitstu1[which(admitstu1[,3] == 'General Arts'),'pgm_rev'] = 'arts'
admitstu1[which(admitstu1[,3] == 'Visual Arts'),'pgm_rev'] = 'arts'
admitstu1[which(admitstu1[,3] == 'Business'),'pgm_rev'] = 'economics'
admitstu1[which(admitstu1[,3] == 'Home Economics'),'pgm_rev'] = 'economics'
admitstu1[which(admitstu1[,3] == 'General Science'),'pgm_rev'] = 'science'
admitstu1[which(is.na(admitstu1[,5])==TRUE),'pgm_rev'] = 'others'
admitstu1[,'choice_rev'] = paste(admitstu1[,'scode_rev'],admitstu1[,'pgm_rev'])
admitstu2 = admitstu1[,c('score','choice_rev')]
#Use aggregate to calculate the cutoff and quality of each choice_rev
choice_rev_cutoff = aggregate(x=admitstu2[,'score'],by=list(admitstu2$choice_rev),FUN=min)
colnames(choice_rev_cutoff)=c('choice_rev','cutoff')
choice_rev_quality = aggregate(x=admitstu2[,'score'],by=list(admitstu2$choice_rev),FUN=mean)
colnames(choice_rev_quality)=c('choice_rev','quality')
choice_rev = merge(admitstu2,choice_rev_cutoff,by='choice_rev',all.x=TRUE)
choice_rev = merge(choice_rev,choice_rev_quality,by='choice_rev',all.x=TRUE)
view(choice_rev)
```

	choice_rev	score	cutoff	quality
1	100 arts	389	361	378.2500
2	100 arts	375	361	378.2500
3	100 arts	403	361	378.2500
4	100 arts	372	361	378.2500
5	100 arts	416	361	378.2500
6	100 arts	375	361	378.2500
7	100 arts	383	361	378.2500
8	100 arts	370	361	378.2500
9	100 arts	413	361	378.2500
10	100 arts	362	361	378.2500

Showing 1 to 10 of 20,000 entries, 4 total columns

#Exercise5==========

#Using the new data with recoded choices, we want to understand the effect of the student test score on his first choice.

#5.1==========

```r
#Propose a model specification. Write the Likelihood function.
choice_rev[,'choice'] = match(choice_rev$choice_rev,sort(unique(choice_rev$choice_rev)))
choice_rev1 = choice_rev[,c('choice','score')]

#Write the likelihood function
like1 = function(param,dat)
{
  score = dat$score
  ch = dat$choice
  ni = nrow(dat)
  nj = length(unique(dat$choice))
  ut = mat.or.vec(ni,nj)
  # multinomial logit
  pn1 = param[1:nj]
  pn2 = param[(nj+1):(2*nj)]
  for (j in 1:nj)
  {
    ut[,j] = score*pn1[j] + pn2[j]
  }
  prob   = exp(ut)            # exp(XB)
  #sprob  = rowsums(prob)      # sum_j exp(XB) denominator
  prob   = sweep(prob,MARGIN=1,FUN="/",STATS=rowSums(prob)) # an example of how to construct
  # match prob to actual choices
  probc = NULL
  for (i in 1:ni)
    {
    probc[i] = prob[i,ch[i]]
    }
  probc[probc>0.999999] = 0.999999
  probc[probc<0.000001] = 0.000001
  like = log(probc)
  return(-sum(like))
}
nj = length(unique(choice_rev$choice))
start1 = runif(2*nj,min=-0.0001,max=0.0001)
opt1 = optim(start1,fn=like1,method="BFGS",control=list(trace=5,REPORT=1,maxit=100),dat=choice_rev1,hessian=FALSE)
```

```
> opt1 = optim(start1,fn=like1,method="BFGS",control=list(trace=5,REPORT=1,maxit=100),dat=choice_rev1,hessian=FALSE)
initial  value 110147.266858
iter   2 value 84053.182484
iter   3 value 82092.170928
iter   4 value 80588.197058
iter   5 value 79330.300937
iter   6 value 78236.169286
iter   7 value 78165.860131
iter   8 value 77702.651771
iter   9 value 77363.420531
iter  10 value 77217.769097

......

iter  80 value 74634.982015
iter  80 value 74634.982015
iter  81 value 74631.758860
iter  82 value 74631.528584
iter  83 value 74631.494762
iter  83 value 74631.494762
iter  84 value 74628.941558
iter  84 value 74628.941558
iter  84 value 74628.941558
final  value 74628.941558
converged
```

#5.2==========

```
> #Estimate parameters and compute the marginal effect of the proposed model.
> opt1$par #estimate parameters
  [1] -2.275991e-02 -2.380966e-02 -2.871357e-02 -2.066166e-02 -1.185236e-02 -1.376238e-02 -1.889690e-02 -1.187508e-02 -1.818093e-02
 [10] -1.855684e-02 -2.484598e-02 -1.956562e-02 -2.745319e-02 -2.807970e-02 -2.866788e-02 -2.679144e-02 -2.393095e-02 -2.461647e-02
 [19] -2.872638e-02 -2.570675e-02 -2.241469e-02 -2.255955e-02 -2.864014e-02 -2.419258e-02 -1.502296e-02 -1.593795e-02 -1.883481e-02
 [28] -1.611654e-02 -2.870534e-02 -2.743901e-02 -2.736098e-02 -2.746194e-02 -1.420913e-02 -1.558159e-02 -2.808310e-02 -1.680989e-02
 [37] -1.755155e-02 -1.944917e-02 -2.292975e-02 -2.090171e-02 -2.683519e-02 -2.679025e-02 -2.742625e-02 -2.739900e-02 -2.351001e-02
 [46] -2.487886e-02 -2.685732e-02 -2.876208e-02 -2.880006e-02 -2.809067e-02 -2.627549e-02 -2.625505e-02 -2.813593e-02 -2.873155e-02
 [55] -2.878222e-02 -1.716161e-02 -1.745657e-02 -2.064667e-02 -1.519955e-02 -1.611885e-02 -1.928134e-02 -1.703330e-02 -2.873610e-02
 [64] -2.876164e-02 -1.930895e-02 -2.051986e-02 -2.318572e-02 -2.155560e-02 -2.363272e-02 -2.873040e-02 -2.678499e-02 -1.161370e-02
 [73] -1.279431e-02 -1.854308e-02 -1.127683e-02 -2.868587e-02 -1.783546e-02 -1.954757e-02 -2.414441e-02 -2.020423e-02 -2.024988e-02
 [82] -2.003959e-02 -2.736460e-02 -2.254616e-02 -2.298985e-02 -2.267397e-02 -2.577562e-02 -2.185394e-02 -2.172377e-02 -2.736321e-02
 [91] -2.267552e-02 -2.873298e-02 -2.800781e-02 -2.689410e-02 -2.730610e-02 -2.226274e-02 -2.136783e-02 -2.871795e-02 -2.225741e-02
[100] -2.493373e-02 -2.574535e-02 -2.807443e-02 -2.435800e-02 -2.872788e-02 -2.799884e-02 -2.877159e-02 -2.874749e-02 -2.739142e-02
[109] -1.691780e-02 -1.766498e-02 -1.913054e-02 -1.660816e-02 -2.744820e-02 -2.804720e-02 -2.877114e-02 -2.793003e-02 -2.806347e-02
[118] -2.747054e-02 -2.872665e-02 -2.878082e-02 -2.874661e-02 -2.879244e-02 -2.801086e-02 -2.876969e-02 -2.573594e-02 -2.806168e-02
[127] -2.867311e-02 -2.746370e-02 -2.397056e-02 -2.527096e-02 -1.120974e-02 -1.247685e-02 -1.901016e-02 -1.128328e-02 -1.564586e-02
[136] -1.725393e-02 -2.335706e-02 -1.784196e-02 -2.111227e-02 -2.361856e-02 -2.678015e-02 -2.394867e-02 -1.784401e-02 -1.931942e-02
[145] -2.435776e-02 -2.092246e-02 -2.464167e-02 -2.578574e-02 -2.487556e-02 -2.813703e-02 -2.575781e-02 -2.803272e-02 -2.100890e-02
[154] -2.335350e-02 -2.680307e-02 -2.337566e-02 -2.346435e-02 -2.434907e-02 -2.467439e-02 -2.333091e-02 -2.879510e-02 -2.875093e-02
[163] -2.814836e-02 -2.683903e-02 -2.874773e-02 -2.876449e-02 -2.359173e-02 -2.428052e-02 -2.743822e-02 -2.746429e-02 -2.743921e-02
[172] -2.872315e-02 -1.732486e-02 -1.948454e-02 -2.090597e-02 -1.676407e-02 -2.874720e-02 -2.743071e-02 -2.527439e-02 -2.872646e-02
[181] -2.869850e-02 -2.871475e-02 -2.467071e-02 -2.874954e-02 -2.875340e-02 -2.457798e-02 -2.741630e-02 -2.740763e-02 -2.879645e-02
[190] -2.677605e-02 -2.868275e-02 -2.873647e-02 -2.808127e-02 -2.880737e-02 -2.347848e-02 -2.418010e-02 -2.571271e-02 -1.757902e-02
[199] -1.808452e-02 -2.413484e-02 -1.833867e-02 -2.744709e-02 -2.880857e-02 -2.873226e-02 -2.812742e-02 -2.876912e-02 -2.295503e-02
[208] -2.154862e-02 -2.748548e-02 -2.087084e-02 -2.131883e-02 -2.129951e-02 -2.678512e-02 -1.935560e-02 -2.749321e-02 -2.616555e-02
[217] -2.410995e-02 -2.879470e-02 -2.570858e-02 -2.868980e-02 -2.808259e-02 -2.806750e-02 -2.806803e-02 -2.798696e-02 -2.873923e-02
[226] -2.877465e-02 -2.869885e-02 -2.620451e-02 -2.747878e-02 -2.042275e-02 -2.280261e-02 -2.485167e-02 -1.926195e-02 -2.808598e-02
[235] -2.869532e-02 -2.737726e-02 -2.318389e-02 -2.527083e-02 -2.526054e-02 -2.743068e-02 -2.360593e-02 -2.486969e-02 -2.418726e-02
[244] -2.155553e-02 -2.735512e-02 -2.871254e-02 -4.381424e-05 -2.121210e-05 -1.007121e-04  5.433381e-05  9.566263e-04  6.094091e-04
[253] -1.230868e-05 -2.965857e-04  5.234493e-05  7.205691e-06  1.925230e-05  6.953003e-05 -9.781419e-05  2.669310e-05 -1.380382e-04
[262] -2.188908e-05 -7.481689e-06 -7.289665e-05 -1.139760e-04 -1.882392e-05 -3.893890e-05  4.312320e-05 -1.175416e-04 -6.425125e-05
[271]  4.161969e-04  1.912232e-04  1.847140e-04  8.324909e-05 -6.672321e-05 -5.166789e-05  1.540585e-05 -9.754880e-06  3.994374e-04
[280]  1.313139e-04  8.605094e-06  2.757193e-05  2.800305e-04  3.755337e-05 -1.024769e-04 -1.039994e-04 -1.568431e-04 -1.556703e-05

[289] -1.300515e-05 -1.041446e-04 -1.246938e-04 -1.515379e-04 -1.421386e-04 -8.143849e-05 -5.283076e-05 -4.853083e-05 -1.396209e-06
[298]  3.889993e-05 -3.642260e-05 -8.264778e-05 -1.224464e-04  7.215860e-05  1.273064e-04  6.189093e-05 -1.453068e-04  1.602905e-04
[307]  5.285920e-05 -1.160149e-04 -4.761787e-05 -2.963416e-05  1.640567e-04  8.856047e-05 -2.566372e-05 -9.487637e-05  2.380667e-05
[316] -1.565696e-04 -8.669005e-05  3.165797e-04  3.356537e-04  1.276642e-04 -1.237202e-03 -1.300974e-04  1.577794e-04  2.435920e-05
[325] -1.397473e-04  5.397894e-05  1.025578e-04 -2.985120e-05  1.655670e-05 -8.020869e-05  3.873957e-05 -4.262915e-05  4.199932e-05
[334]  7.294839e-05 -3.836793e-05 -1.597071e-05 -1.010994e-04 -8.151921e-05 -1.673074e-04 -4.682863e-05 -1.187908e-04 -2.296465e-05
[343] -5.205680e-05 -1.379222e-04 -1.943706e-05  2.413893e-05 -6.448575e-05 -1.018156e-04 -1.183101e-04 -1.686675e-04 -1.287734e-04
[352] -3.565183e-05 -1.188696e-04 -1.246699e-04  3.125799e-04  9.700310e-05 -3.987120e-05  1.248928e-04 -1.888836e-05  4.586224e-06
[361] -1.342477e-04 -3.344270e-05 -5.263945e-05 -1.429265e-04 -5.444195e-05 -4.575166e-05 -5.361675e-05 -4.578291e-05 -1.529168e-04
[370] -1.160860e-04 -4.042330e-05 -1.194335e-04 -1.518406e-04 -1.093567e-04  2.670356e-06 -1.186725e-04  2.573931e-03  1.275216e-03
[379]  1.703407e-05  8.101347e-04  3.613344e-04  2.261485e-04 -5.762281e-05 -4.549231e-05 -4.315407e-05 -5.343166e-05  3.499594e-05
[388]  7.242977e-06  1.450814e-04  1.343361e-04 -1.312207e-04  6.993380e-05 -3.655589e-05  1.909660e-05 -1.294825e-04 -5.586115e-05
[397] -1.431090e-04 -1.329695e-04 -8.518360e-06  6.739863e-05 -7.803743e-05  6.096074e-05 -8.919782e-06 -2.035579e-05  4.193388e-05
[406] -5.959299e-05 -7.438696e-06  1.922705e-05 -6.899826e-05  1.132130e-05 -7.810386e-05 -2.818889e-05 -4.970433e-05  6.923351e-06
[415] -1.197820e-04 -6.212601e-05 -1.072212e-04 -1.225287e-04  2.446182e-04  4.826686e-05  6.718124e-05  1.916767e-04 -2.047179e-05
[424]  1.291491e-05  2.100948e-05 -9.048515e-06 -9.765013e-05 -2.940099e-05 -5.288169e-05 -2.212677e-05 -1.058082e-04 -1.095850e-04
[433] -1.421770e-04  9.492316e-06 -9.321814e-05 -1.347794e-04  1.946011e-05 -1.700615e-04 -1.210032e-04 -1.336083e-04 -5.627361e-05
[442] -1.358466e-04 -3.611545e-06  1.746816e-04  4.393706e-05  3.960721e-05  1.332122e-04 -7.990112e-05 -9.009496e-05 -1.152637e-04
[451] -1.680453e-04  2.427313e-05 -6.118973e-05 -5.734385e-05 -1.351294e-04  9.793317e-05  1.927390e-05 -6.433270e-05  3.285277e-05
[460]  1.783109e-05 -5.433601e-05 -1.289493e-04 -7.396032e-05 -1.577144e-04 -2.266358e-05 -3.531056e-05 -2.998286e-05 -1.251725e-04
[469] -1.396941e-05 -1.400841e-04 -1.292977e-04  1.345930e-05 -9.981709e-06 -6.454935e-05 -4.150552e-05  1.738494e-05  1.693181e-05
[478] -7.218162e-05 -1.679812e-05 -2.868962e-06 -1.372391e-05 -8.381236e-05 -1.038741e-04  3.096896e-05  2.703854e-06 -1.239474e-04
[487]  2.622480e-05  2.516034e-05 -9.970231e-05 -2.626631e-05 -5.203877e-05 -6.190453e-05
```

```
#Compute the marginal effect
beta1 = mat.or.vec(2,nj)
beta1[1,2:246] = coef1[1:245,1]
beta1[2,2:246] = coef1[1:245,2]
X1 = cbind(rep(1,nrow(choice_rev1)),choice_rev1$score)
Y_hat1 = X1 %*% beta1
pr1 = mat.or.vec(20000,246)
for (i in 1:246) {
  pr1[,i] = dnorm(Y_hat1[,i])
}
ME1 = as.data.frame(apply(pr1,2,mean))
ME1 = cbind(unique(choice_rev$choice_rev),ME1)
colnames(ME1) = c('choice','ME')
view(ME1)
```

	choice	ME
1	100 arts	3.989423e-01
2	100 economics	3.983233e-01
3	100 others	2.521146e-01
4	100 science	2.352833e-01
5	101 arts	1.453139e-04
6	101 economics	6.725379e-03
7	101 others	3.271556e-02
8	101 science	3.485642e-03
9	102 arts	7.038918e-02
10	102 economics	1.142160e-01
11	102 others	3.987225e-01
12	102 science	1.797295e-01
13	103 arts	2.053174e-01
14	103 economics	2.760585e-01
15	103 others	3.728736e-01
16	103 science	3.363918e-01
17	104 arts	3.577661e-01

Showing 1 to 18 of 246 entries, 2 total columns

#Exercise6==========

#Using the new data with recoded choices, we want to understand the effect of the school quality on the first choice.

#6.1==========

```
#Propose a model specification. Write the Likelihood function.
choice_rev2 = choice_rev[,c('choice','quality')]
like2 = function(param,dat)
{
  quality = dat$quality
  ch = dat$choice
  ni = nrow(dat)
  nj = length(unique(dat$choice))
  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)) # an example of how to construct
  # 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))
  return(-like)
}
start2 = runif(2*nj,min=-0.0001,max=0.0001)
opt2 = optim(start2,fn=like2,method="BFGS",control=list(trace=5,REPORT=1,maxit=100),dat=choice_rev2,hessian=FALSE)
```

```
> opt2 = optim(start2,fn=like2,method="BFGS",control=list(trace=5,REPORT=1,maxit=100),dat=choice_rev2,hessian=FALSE)
initial  value 109976.819635
iter   2 value 83925.466936
iter   3 value 81992.377133
iter   4 value 80523.523822
iter   5 value 79245.486486
iter   6 value 78170.451343
iter   7 value 78167.215476
iter   8 value 77722.368084
iter   9 value 77674.167212
iter  10 value 77539.798117

......
iter  88 value 74602.219026
iter  89 value 74599.293508
iter  90 value 74599.244163
iter  90 value 74599.244163
iter  91 value 74584.779908
iter  92 value 74583.943223
iter  93 value 74583.208213
iter  93 value 74583.208213
iter  94 value 74570.284530
iter  94 value 74570.284530
iter  94 value 74570.284530
final  value 74570.284530
converged
```

#6.2==========

```
> #Estimate parameters and compute marginal effect of the proposed model.
> opt2$par
  [1] -2.838089e-02 -2.961609e-02 -3.476822e-02 -2.657851e-02 -1.768166e-02 -1.954589e-02 -2.475638e-02 -1.769788e-02 -2.406144e-02
 [10] -2.439436e-02 -3.066958e-02 -2.553715e-02 -3.302569e-02 -3.376078e-02 -3.442310e-02 -3.222877e-02 -2.976871e-02 -3.046471e-02
 [19] -3.457310e-02 -3.129167e-02 -2.808395e-02 -2.820526e-02 -3.464107e-02 -3.005160e-02 -2.085896e-02 -2.177845e-02 -2.468919e-02
 [28] -2.196013e-02 -3.455790e-02 -3.294857e-02 -3.286092e-02 -3.294443e-02 -1.996639e-02 -2.143146e-02 -3.378675e-02 -2.264231e-02
 [37] -2.338262e-02 -2.537403e-02 -2.855750e-02 -2.685075e-02 -3.226367e-02 -3.224942e-02 -3.298472e-02 -3.298152e-02 -2.925176e-02
 [46] -3.069235e-02 -3.231526e-02 -3.472754e-02 -3.467855e-02 -3.380048e-02 -3.175353e-02 -3.171495e-02 -3.376490e-02 -3.468335e-02
 [55] -3.451062e-02 -2.297553e-02 -2.329799e-02 -2.656146e-02 -2.105087e-02 -2.196289e-02 -2.517212e-02 -2.285931e-02 -3.467124e-02
 [64] -3.465510e-02 -2.520496e-02 -2.642611e-02 -2.885315e-02 -2.748029e-02 -2.941005e-02 -3.451321e-02 -3.224032e-02 -1.743125e-02
 [73] -1.859456e-02 -2.438028e-02 -1.708402e-02 -3.465711e-02 -2.371158e-02 -2.551314e-02 -2.999708e-02 -2.611965e-02 -2.616510e-02
 [82] -2.596345e-02 -3.289648e-02 -2.818922e-02 -2.862361e-02 -2.829655e-02 -3.133891e-02 -2.769827e-02 -2.760511e-02 -3.288021e-02
 [91] -2.830362e-02 -3.466063e-02 -3.369241e-02 -3.230168e-02 -3.277854e-02 -2.797375e-02 -2.733170e-02 -3.462428e-02 -2.796799e-02
[100] -3.072656e-02 -3.132697e-02 -3.378029e-02 -3.023233e-02 -3.473221e-02 -3.380066e-02 -3.464284e-02 -3.465084e-02 -3.297185e-02
[109] -2.273825e-02 -2.353221e-02 -2.500132e-02 -2.244860e-02 -3.296258e-02 -3.374495e-02 -3.451686e-02 -3.364801e-02 -3.371924e-02
[118] -3.298419e-02 -3.452290e-02 -3.451081e-02 -3.463862e-02 -3.457557e-02 -3.365811e-02 -3.455929e-02 -3.132287e-02 -3.381667e-02
[127] -3.469471e-02 -3.300838e-02 -2.979188e-02 -3.099155e-02 -1.701235e-02 -1.826668e-02 -2.487511e-02 -1.708128e-02 -2.147540e-02
[136] -2.306387e-02 -2.907625e-02 -2.371683e-02 -2.708798e-02 -2.939757e-02 -3.226471e-02 -2.977889e-02 -2.371875e-02 -2.521721e-02
[145] -3.022163e-02 -2.687557e-02 -3.048962e-02 -3.135807e-02 -3.069018e-02 -3.384180e-02 -3.133872e-02 -3.377052e-02 -2.697305e-02
[154] -2.906173e-02 -3.224803e-02 -2.909713e-02 -2.920305e-02 -3.021207e-02 -3.052278e-02 -2.903561e-02 -3.468114e-02 -3.469080e-02
[163] -3.380474e-02 -3.223536e-02 -3.470130e-02 -3.452130e-02 -2.936766e-02 -3.014587e-02 -3.291876e-02 -3.304264e-02 -3.301495e-02
[172] -3.476088e-02 -2.313689e-02 -2.542039e-02 -2.685664e-02 -2.259526e-02 -3.458823e-02 -3.304227e-02 -3.098202e-02 -3.451609e-02
[181] -3.473640e-02 -3.450390e-02 -3.052352e-02 -3.466835e-02 -3.466013e-02 -3.043519e-02 -3.300733e-02 -3.290912e-02 -3.478290e-02
[190] -3.219225e-02 -3.459514e-02 -3.474384e-02 -3.383073e-02 -3.467824e-02 -2.922622e-02 -3.003582e-02 -3.130538e-02 -2.341630e-02
[199] -2.396491e-02 -2.999230e-02 -2.419868e-02 -3.297941e-02 -3.450855e-02 -3.473564e-02 -3.373304e-02 -3.464687e-02 -2.859143e-02
[208] -2.747868e-02 -3.300739e-02 -2.681566e-02 -2.729086e-02 -2.726837e-02 -3.226801e-02 -2.526138e-02 -3.299976e-02 -3.166831e-02
[217] -2.996663e-02 -3.453582e-02 -3.128545e-02 -3.468169e-02 -3.386439e-02 -3.374057e-02 -3.371636e-02 -3.369034e-02 -3.475553e-02
[226] -3.471440e-02 -3.459493e-02 -3.168334e-02 -3.297408e-02 -2.632948e-02 -2.841531e-02 -3.066901e-02 -2.514897e-02 -3.385444e-02
[235] -3.467609e-02 -3.283770e-02 -2.885669e-02 -3.098047e-02 -3.097812e-02 -3.290728e-02 -2.937239e-02 -3.067980e-02 -3.004260e-02
[244] -2.748186e-02 -3.293666e-02 -3.469844e-02  3.482444e-05 -4.737469e-06 -6.676054e-05  1.626146e-05  1.056391e-03  7.712386e-04
[253] -4.073583e-05 -6.207361e-04  7.606873e-05  2.011299e-04 -1.019590e-04  9.716757e-05 -3.467351e-05 -4.272635e-05 -6.015243e-05
[262] -7.920608e-05 -1.367649e-04 -1.428184e-04 -1.031284e-04 -1.485337e-04 -1.082359e-04  8.785881e-05 -8.857345e-05 -5.317187e-05
```

```
[271]  5.188929e-04  3.199009e-04  2.213556e-04  1.981568e-04 -1.465270e-04 -1.027905e-04 -3.861486e-05 -1.673086e-04  5.644989e-04
[280]  1.551495e-04 -1.568632e-04  5.957642e-05  1.719378e-04  1.453599e-04 -1.015791e-04 -2.139440e-05 -3.383498e-05 -8.524997e-05
[289] -1.227844e-04 -8.649936e-06  5.291255e-05 -9.476420e-05 -3.678325e-05 -5.850701e-05 -9.741094e-06  8.914326e-06 -3.797479e-05
[298]  1.046464e-05 -1.681151e-04 -1.546061e-04 -1.595791e-05  1.177790e-04  3.887775e-05  4.711380e-05 -1.656247e-04  7.106488e-06
[307]  1.000497e-04 -9.482762e-05 -7.277031e-05 -7.427215e-05  2.131146e-04 -5.324300e-05  4.372299e-05  5.673048e-05 -4.354509e-05
[316] -3.096689e-05  2.764918e-05  3.132717e-04  1.811519e-04  1.846206e-05 -1.772998e-03 -9.105956e-05  2.432556e-04  2.537313e-05
[325] -9.057868e-06 -3.060764e-05  1.718478e-05  1.271217e-04 -1.489383e-04 -9.545689e-05  1.094778e-05  2.487898e-06  2.633301e-05
[334]  8.774262e-05  8.260333e-05 -1.464971e-04  4.521626e-06  4.494071e-06 -5.044095e-05  2.281021e-05 -1.737628e-04 -4.871998e-05
[343]  3.692771e-06 -2.988046e-05 -1.334228e-05 -1.903039e-05 -3.982186e-05 -1.567852e-05 -5.551836e-05 -1.149086e-04 -1.061375e-04
[352] -1.581456e-05 -4.812258e-05 -5.028165e-05  3.879270e-04  3.254839e-04  1.034145e-04  2.381325e-04 -1.478039e-05 -1.639729e-05
[361] -2.552410e-05 -4.868279e-05 -1.647453e-04 -4.398460e-05 -1.725984e-05 -1.352404e-04 -1.285128e-04 -1.169976e-04 -1.454462e-04
[370] -1.832389e-04 -1.225698e-04  4.839918e-06 -1.692239e-04 -3.659370e-05 -1.144734e-04 -8.653533e-05  2.681440e-03  1.515677e-03
[379]  1.571394e-04  6.451178e-04  4.386897e-04  1.125095e-04 -3.863214e-05  5.832407e-05  1.304969e-05 -1.245407e-04 -3.807759e-05
[388] -4.179129e-05  1.805632e-04  1.053279e-04 -7.313453e-05  9.975291e-05 -5.314741e-05 -1.772275e-05 -4.583930e-05 -1.628751e-04
[397] -1.508135e-04  8.200234e-06  9.647465e-07  8.807262e-06 -1.129454e-05  1.268463e-05  5.691034e-05 -1.150663e-04 -1.214787e-04
[406] -1.109123e-04 -3.491461e-05  4.559196e-06 -2.066068e-05 -1.182103e-05 -1.305943e-04 -3.978110e-05 -7.099949e-05 -7.179726e-05
[415]  8.241346e-06 -3.856343e-05  1.747454e-05 -1.741805e-04  2.955160e-04  2.483073e-05 -1.286477e-05  1.638918e-04 -1.392347e-05
[424] -4.842272e-06 -1.969552e-05  3.430620e-06 -6.575408e-05 -1.114687e-04  5.715408e-06 -1.765692e-04 -1.478409e-04 -1.252008e-04
[433] -4.740837e-05 -1.493854e-04 -1.831996e-04 -5.282356e-05 -1.819475e-04 -1.360396e-04 -8.186440e-05  6.461734e-06  3.211662e-05
[442] -9.966963e-05 -1.322519e-04  2.311406e-04  1.084026e-04 -7.827110e-05  1.329811e-04 -9.136471e-05 -9.798776e-05 -4.289967e-05
[451] -1.240023e-04 -1.612721e-04  2.552135e-05  5.636249e-06 -8.106433e-05 -3.647661e-05  3.962046e-05 -4.835973e-06 -1.081492e-04
[460]  1.623572e-04 -4.296638e-05  7.269231e-06 -8.998379e-05 -1.821444e-04 -1.342201e-04 -1.667229e-05 -2.433660e-06 -1.803530e-04
[469] -4.652655e-05 -2.842739e-05 -1.214008e-04 -1.258061e-05 -8.749738e-05  1.599883e-06 -1.602220e-04  5.370853e-05  5.020775e-05
[478] -5.487047e-05  1.229947e-04 -2.643511e-05 -7.579110e-05 -8.039723e-05  2.205643e-05 -9.127620e-05 -1.068174e-04 -3.282270e-05
[487] -5.041573e-05 -6.417997e-05 -8.940888e-05  5.812521e-05 -2.137801e-05 -1.634693e-04
```

```r
#Compute the marginal effect
beta2 = mat.or.vec(2,nj)
beta2[1,2:246] = coef2[1:245,1]
beta2[2,2:246] = coef2[1:245,2]
X2 = cbind(rep(1,nrow(choice_rev2)),choice_rev2$quality)
Y_hat2 = X2 %*% beta2
pr2 = mat.or.vec(20000,246)
for (i in 1:246) {
  pr2[,i] = dnorm(Y_hat2[,i])
}
ME2 = as.data.frame(apply(pr2,2,mean))
ME2 = cbind(unique(choice_rev$choice_rev),ME2)
colnames(ME2) = c('choice','ME')
view(ME2)
```

	choice	ME
1	100 arts	3.989423e-01
2	100 economics	3.620315e-01
3	100 others	3.349213e-01
4	100 science	3.104597e-01
5	101 arts	3.585164e-05
6	101 economics	8.433589e-05
7	101 others	2.060381e-03
8	101 science	7.118937e-03
9	102 arts	6.841297e-02
10	102 economics	5.601443e-02
11	102 others	3.275512e-01
12	102 science	2.194555e-01
13	103 arts	1.573793e-02
14	103 economics	3.962579e-01
15	103 others	8.772383e-02
16	103 science	3.563400e-01
17	104 arts	1.570501e-01

Showing 1 to 18 of 246 entries, 2 total columns

#Exercise7==========

#7.1==========

```r
#Explain and justify, which model (first or second model) you think is appropriate to conduct this exercise.
#Delete the rows whose program is "others"
admitstu3 = admitstu1[-which(admitstu1$pgm_rev == 'others'),]
admitstu3 = admitstu3[,c('score','choice_rev')]
#Use aggregate to calculate the cutoff and quality of each choice_rev
choice_rev_cutoff1 = aggregate(x=admitstu3['score'],by=list(admitstu3$choice_rev),FUN=min)
colnames(choice_rev_cutoff1)=c('choice_rev','cutoff')
choice_rev_quality1 = aggregate(x=admitstu3['score'],by=list(admitstu3$choice_rev),FUN=mean)
colnames(choice_rev_quality1)=c('choice_rev','quality')
choice_rev_delete = merge(admitstu3,choice_rev_cutoff1,by='choice_rev',all.x=TRUE)
choice_rev_delete = merge(choice_rev_delete,choice_rev_quality1,by='choice_rev',all.x=TRUE)
view(choice_rev_delete)

#Run the first model: choice~score
choice_rev_delete['choice'] = match(choice_rev_delete$choice_rev,sort(unique(choice_rev_del$choice_rev)))
choice_rev3 = choice_rev_delete[,c('choice','score')]
nj_3 = length(unique(choice_rev3$choice))
start3 = runif(2*nj_3,min=-0.0001,max=0.0001)
opt1_delete = optim(start3,fn=like1,method="BFGS",control=list(trace=5,REPORT=1,maxit=100),dat=choice_rev3,hessian=FALSE)
```

```
> opt1_delete = optim(start3,fn=like1,method="BFGS",control=list(trace=5,REPORT=1,maxit=100),dat=choice_rev3,hessian=FALSE)
initial  value 101642.361301
iter   2 value 78208.336477
iter   3 value 76331.970780
iter   4 value 74959.099524
iter   5 value 73801.037473
iter   6 value 72813.910758
iter   7 value 72280.727385
iter   8 value 71882.700115
iter   9 value 71618.191129
iter  10 value 71428.200933
......
iter  70 value 69605.577066
iter  71 value 69605.553065
iter  71 value 69605.553065
iter  72 value 69602.097752
iter  73 value 69601.898572
iter  73 value 69601.898572
iter  74 value 69601.517615
iter  74 value 69601.517615
iter  74 value 69601.517615
final  value 69601.517615
converged
```

```r
#Run the second model: choice~quality
choice_rev4 = choice_rev_delete[,c('choice','quality')]
nj_4 = length(unique(choice_rev4$choice))
star4 = runif(2*nj_4,min=-0.0001,max=0.0001)
opt2_delete = optim(start4,fn=like2,method="BFGS",control=list(trace=5,REPORT=1,maxit=100),dat=choice_rev4,hessian=FALSE)
```

```
initial  value 101721.434484
iter  10 value 71225.456076
iter  20 value 71001.852878
iter  30 value 71000.844187
iter  40 value 71000.342855
iter  50 value 70975.480919
iter  60 value 70586.490846
iter  70 value 64707.570713
iter  80 value 64550.551545
iter  90 value 64548.283175
iter 100 value 64120.000426
final  value 64120.000426
```

#The second model is better, because the return likelihood is smaller.

#7.2==========

```r
#Calculate choice probabilities under the appropriate model.
beta4 = mat.or.vec(2,nj_4)
beta4[1,2:195] = coef4[1:194,1]
beta4[2,2:195] = coef4[1:194,2]
X4 = cbind(rep(1,nrow(choice_rev4)),choice_rev4$quality)
Y_hat4 = X4 %*% beta4
pr4 = mat.or.vec(nrow(Y_hat4),195)
for (i in 1:195) {
  pr4[,i] = dnorm(Y_hat4[,i])
}
view(pr4)
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13
1	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
2	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
3	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
4	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
5	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
6	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
7	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
8	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
9	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
10	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
11	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
12	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
13	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
14	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
15	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
16	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087
17	0.3989423	0.03113750	0.2660305	9.549289e-05	0.0006231608	1.630981e-03	0.07660869	0.10263124	0.3495779	0.2055923	0.3587798	0.02028319	0.3943087

Showing 1 to 17 of 1,000 entries, 195 total columns