HW3_613_Sun

Mike Sun

3/22/2022

```
dw = "~/Desktop/HW3/Data/"

datjss <- read.csv(paste(dw,"datjss.csv",sep=""))
datsss <- read.csv(paste(dw,"datsss.csv",sep=""))
datstu_v2 <- read.csv(paste(dw,"datstu_v2.csv",sep=""))

datsss_nna = na.omit(datsss)
```

*Exercise*1

Q1: Number of students, schools, programs

Students:

```
student_num = length(datstu_v2$V1)
student_num
```

```
## [1] 340823
```

Schools:

```
school_num = length(unique(datsss_nna$schoolcode))
school_num
```

```
## [1] 689
```

Programs:

```
program <- subset(datstu_v2,select =
                        c(choicepgm1,choicepgm2,choicepgm3,choicepgm4,choicepgm5,choicepgm6))
program <- program %>%
  pivot_longer(cols = starts_with("choicepgm"),names_to = "program_rank",values_to = "choiceprogram")

program_num = length(unique(program$choiceprogram))

program_num
```

```
## [1] 33
```

Q2: Number of choices (school, program)

```
school <- subset(datstu_v2,select =
                        c(V1,schoolcode1,schoolcode2,schoolcode3,schoolcode4,schoolcode5,schoolcode6))
program <- subset(datstu_v2,select =
                        c(choicepgm1,choicepgm2,choicepgm3,choicepgm4,choicepgm5,choicepgm6))

school <- school %>%
```

```
  pivot_longer(cols = starts_with("schoolcode"),names_to = "school_rank",values_to = "schoolcode")

program <- program %>%
  pivot_longer(cols = starts_with("choicepgm"),names_to = "program_rank",values_to = "choiceprogram")

school_program_matrix <- cbind(school,program) %>% select(schoolcode,choiceprogram) %>% unique()
```

```
length(school_program_matrix$schoolcode)
```

```
## [1] 3086
```

Q3: Number of students applying to at least one senior high schools in the same district to home:

```
school_district <- subset(datstu_v2,select =
                          c(V1,schoolcode1,schoolcode2,schoolcode3,schoolcode4,
                            schoolcode5,schoolcode6,jssdistrict))

school_district <- school_district %>% pivot_longer(
  cols = starts_with("schoolcode"),names_to = "schoolrank",values_to = "schoolcode")

district = na.omit(datsss_nna %>% select(schoolcode,sssdistrict) %>% unique())

school_district = left_join(school_district,district,by = "schoolcode")
```

```
district_check = school_district %>%
  mutate(district_same = ifelse(jssdistrict == sssdistrict,1,0)) %>%
  select(V1,district_same) %>%
  unique()
```

```
apply_sch_district = subset(district_check, district_same == 1)
length(apply_sch_district$V1)
```

```
## [1] 262167
```

Q4,5,6: Number of students each senior high school admitted, the cutoff of senior high schools, and the quality of senior high schools:

```
school_rs <- subset(datstu_v2,select =
                          c(V1,schoolcode1,schoolcode2,schoolcode3,schoolcode4,
                            schoolcode5,schoolcode6,score,rankplace))

school_rs = subset(subset(school_rs,is.na(rankplace) == FALSE),is.na(score) == FALSE)

school_rs <- school_rs %>% pivot_longer(
  cols = starts_with("schoolcode"),names_to = "schoolrank",values_to = "schoolcode")

school_rs <- school_rs %>%
  mutate(schoolrank_num = ifelse(schoolrank == "schoolcode1",1,
                                 ifelse(schoolrank == "schoolcode2",2,
                                        ifelse(schoolrank == "schoolcode3",3,
                                               ifelse(schoolrank == "schoolcode4",4,
                                                      ifelse(schoolrank == "schoolcode5",5,6))))))

school_rs$rankplace = as.numeric(school_rs$rankplace)
school_rs$score = as.numeric(school_rs$score)
```

```r
school_rs_match = subset(school_rs,rankplace == schoolrank_num)

## admission
school_rs_final = school_rs_match %>% group_by(schoolcode) %>%
  mutate(school_admission_count = n()) %>%
  mutate(ave_score = mean(score)) %>%
  mutate(min_score = min(score)) %>%
  select(schoolcode,school_admission_count,min_score,ave_score) %>%
  unique()

head(school_rs_final)
```

```
## # A tibble: 6 x 4
## # Groups:   schoolcode [6]
##   schoolcode school_admission_count min_score ave_score
##        <int>                  <int>     <dbl>     <dbl>
## 1      30403                     63       208      244.
## 2      21001                    449       252      297.
## 3    9021002                     56       204      247.
## 4      70503                    255       205      245.
## 5      21303                    462       312      343.
## 6      30402                    530       192      250.
```

Exercise2

Create a school level dataset, where each row corresponds to a (school,program) with the following variables:

Q1&2: The district where the school is located, and the latitude and longitude of the district:

```
datsss_nna_new = subset(datsss_nna,select=
                          c(schoolcode,sssdistrict, ssslong, ssslat)) %>% unique()

sch_prog = left_join(unique(school_program_matrix),datsss_nna_new,by="schoolcode")
```

Q3,4,5: cutoff (the lowest score to be admitted), quality (the average score of the students admitted), and size (number of students admitted)

```
school_matrix_full <- datstu_v2 %>%
  pivot_longer(cols = starts_with("schoolcode"),names_to = "school_rank",values_to = "schoolcode")

program_matrix_full <- datstu_v2 %>%
  pivot_longer(cols = starts_with("choicepgm"),names_to = "program_rank",values_to = "choiceprogram") %
  select(program_rank,choiceprogram)

sp_bind = cbind(school_matrix_full,program_matrix_full)
sp_full = subset(sp_bind,select =
  -c(choicepgm1,choicepgm2,choicepgm3,choicepgm4,choicepgm5,choicepgm6))
```

```
sp_rs = subset(sp_full,select =
                c(score,rankplace,school_rank,program_rank,schoolcode,choiceprogram))

sp_rs = subset(subset(sp_rs,is.na(rankplace) == FALSE),is.na(score) == FALSE)

sp_rs <- sp_rs %>%
  mutate(schoolrank_num = ifelse(school_rank == "schoolcode1",1,
                                 ifelse(school_rank == "schoolcode2",2,
                                        ifelse(school_rank == "schoolcode3",3,
                                               ifelse(school_rank == "schoolcode4",4,
                                                      ifelse(school_rank == "schoolcode5",5,6)))))) %>%
  mutate(programrank_num = ifelse(program_rank == "choicepgm1",1,
                                 ifelse(program_rank == "choicepgm2",2,
                                        ifelse(program_rank == "choicepgm3",3,
                                               ifelse(program_rank == "choicepgm4",4,
                                                      ifelse(program_rank == "choicepgm5",5,6))))))

sp_rs$rankplace = as.numeric(sp_rs$rankplace)
sp_rs$score = as.numeric(sp_rs$score)
```

```
sp_rs_match = subset(sp_rs,rankplace == schoolrank_num)

## admission
sp_rs_final = sp_rs_match %>% group_by(schoolcode,choiceprogram) %>%
  mutate(school_admission_count = n()) %>%
  mutate(ave_score = mean(score)) %>%
  mutate(min_score = min(score)) %>%
  select(choiceprogram,schoolcode,school_admission_count,min_score,ave_score) %>%
  unique()

head(sp_rs_final)
```

```
## # A tibble: 6 x 5
## # Groups:   schoolcode, choiceprogram [6]
##   choiceprogram   schoolcode school_admission_count min_score ave_score
##   <chr>                <int>                  <int>     <dbl>     <dbl>
## 1 General Arts         30403                     38       208      245.
## 2 Agriculture          30403                     15       219      242.
## 3 Home Economics       30403                      8       215      248.
## 4 Business             30403                      2       227      233 
## 5 General Science      21001                     62       252      293.
## 6 General Arts         21001                    180       276      302.
```

Exercise3

Distance:

```
distance_school = left_join(sp_full, datsss_nna_new,by="schoolcode")
datjss_clean = subset(datjss,select = -c(X))
distance_school = left_join(distance_school,datjss_clean,by="jssdistrict")

distance_school = distance_school %>%
  mutate(distance =
           sqrt((69.172*(ssslong-point_x)*cos(point_y/57.3))^2+(69.172*(ssslat-point_y))^2)
         )
```

```
head(distance_school$distance)
```

```
## [1]  8.813579  8.813579 18.895053 18.895053 17.179653 63.917746
```

Exercise4

Q1: Recode the schoolcode into its first three digits

```
sp_full = sp_full %>%
  mutate(scode_rev = substr(sp_full$schoolcode,1,3))

sp_full$scode_rev = as.numeric(sp_full$scode_rev)
```

Q2: Recode the program variable into 4 categories: arts (general arts and visual arts), economics (business and home economics), science (general science) and others

```
sp_full = sp_full %>%
  mutate(pgm_rev =
           ifelse(choiceprogram == "General Arts","arts",
                ifelse(choiceprogram == "Visual Arts","arts",
                    ifelse(choiceprogram == "Business","economics",
                           ifelse(choiceprogram == "Home Economics","economics",
                                  ifelse(choiceprogram == "General Science","science","others"))))))
```

Q3: Create a new choice variable choice rev

```
sp_full = sp_full %>%
  mutate(choice_rev = paste(scode_rev,pgm_rev,sep=""))
```

Q4: Recalculate the cutoff and the quality for each recoded choice

```
sp_full <- sp_full %>%
  mutate(schoolrank_num = ifelse(school_rank == "schoolcode1",1,
                                 ifelse(school_rank == "schoolcode2",2,
                                        ifelse(school_rank == "schoolcode3",3,
                                               ifelse(school_rank == "schoolcode4",4,
                                                      ifelse(school_rank == "schoolcode5",5,6)))))) %>%
  mutate(programrank_num = ifelse(program_rank == "choicepgm1",1,
                                 ifelse(program_rank == "choicepgm2",2,
                                        ifelse(program_rank == "choicepgm3",3,
                                               ifelse(program_rank == "choicepgm4",4,
                                                      ifelse(program_rank == "choicepgm5",5,6))))))

sp_full$rankplace = as.numeric(sp_full$rankplace)
sp_full$score = as.numeric(sp_full$score)
```

```
sp_full_match = subset(sp_full,is.na(rank)==FALSE)
```

```
## Warning in is.na(rank): is.na() applied to non-(list or vector) of type
## 'closure'
```

```
sp_full_match = subset(sp_full,rankplace == schoolrank_num)


## cutoff and quality
sp_full_final = sp_full_match %>% group_by(choice_rev) %>%
  mutate(ave_score = mean(score)) %>%
  mutate(min_score = min(score)) %>%
  select(V1,choice_rev,min_score,ave_score)

sp_full_unique = sp_full_final %>% select(choice_rev,min_score,ave_score) %>% unique()
```

```
head(sp_full_unique)
```

```
## # A tibble: 6 x 3
## # Groups:   choice_rev [6]
##   choice_rev   min_score ave_score
##   <chr>            <dbl>     <dbl>
## 1 304arts            207      295.
## 2 304others          219      319.
## 3 304economics       192      298.
## 4 210science         206      333.
## 5 210arts            208      291.
## 6 210economics       203      294.
```

Q5: Consider the 20,000 highest score students

```
high_score <- unique(subset(sp_full, select=c(V1,score)))
```

```
topscore =arrange(high_score,desc(score))
topscore = topscore$V1[1:20000]

top_sp = sp_full %>% filter(V1%in%topscore)

top_sp_full = left_join(top_sp,sp_full_unique,by=c("choice_rev"))

head(top_sp_full)
```

```
##       V1 score agey male       jssdistrict rankplace school_rank schoolcode
## 1 179982   375   17    0 Ga East (Abokobi)         1 schoolcode1      21001
## 2 179982   375   17    0 Ga East (Abokobi)         1 schoolcode2      21002
## 3 179982   375   17    0 Ga East (Abokobi)         1 schoolcode3      21006
## 4 179982   375   17    0 Ga East (Abokobi)         1 schoolcode4      21009
## 5 179982   375   17    0 Ga East (Abokobi)         1 schoolcode5      21401
## 6 179982   375   17    0 Ga East (Abokobi)         1 schoolcode6      21201
##   program_rank  choiceprogram scode_rev   pgm_rev   choice_rev schoolrank_num
## 1   choicepgm1       Business       210 economics 210economics              1
## 2   choicepgm2       Business       210 economics 210economics              2
## 3   choicepgm3       Business       210 economics 210economics              3
## 4   choicepgm4       Business       210 economics 210economics              4
## 5   choicepgm5 Home Economics       214 economics 214economics              5
## 6   choicepgm6 Home Economics       212 economics 212economics              6
##   programrank_num min_score ave_score
## 1               1       203  294.2891
## 2               2       203  294.2891
## 3               3       203  294.2891
## 4               4       203  294.2891
## 5               5       207  267.6195
## 6               6       213  264.5061
```

*Exercise*5

Propose a model specification. Write the Likelihood function:

I believe conditional logit serves better since we want to explore scores effect on each of school-program choice's probability instead of relative probability of other programs to one program (mlogit concept):

```
top_sp_first <- top_sp_full %>% filter(schoolrank_num== 1) %>%
  mutate(choice = factor(choice_rev))
```

```
like_fun = function(param)
{
  data = top_sp_first
  score =  data$score
  choice = data$choice

  ni = nrow(data)
  nj = length(unique(choice))
  ut = mat.or.vec(ni,nj)

  for (j in 1:nj)
  {
    # conditional logit
    ut[,j] = param[1] + param[2]*score[j]
  }
  prob   = exp(ut)
  prob   = sweep(prob,MARGIN=1,FUN="/",STATS=rowSums(prob))
  probc = prob[,1]

  probc[probc>0.999999] = 0.999999
  probc[probc<0.000001] = 0.000001
  like = sum(log(probc))
  return(-like)
}
```

testing:

```
npar = 2
param = runif(npar)
like_fun(param)
```

```
## [1] 276310.2
```

Optimization:

```
optim(par = c(-0.5,-0.2), fn=like_fun,method="BFGS")$par
```

```
## [1] -0.500000 -0.021382
```

By optimization results, we have beta equals -0.02. I fail to compute the marginal effect of beta obtained here.

*Exercise*6

Propose a model specification. Write the Likelihood function:

Again, conditional logit will help us explore the relationship better, since we want to explore school quality (average admission effect) on each of school-program choice's probability instead of relative probability of other programs to one program:

```
like_fun_2 = function(param)
{
  data = top_sp_first
  ave_score =  data$ave_score
  choice = data$choice

  ni = nrow(data)
  nj = length(unique(choice))
  ut = mat.or.vec(ni,nj)

  for (j in 1:nj)
  {
    # conditional logit
    ut[,j] = param[1] + param[2]*ave_score[j]
  }
  prob   = exp(ut)
  prob   = sweep(prob,MARGIN=1,FUN="/",STATS=rowSums(prob))
  probc = prob[,1]

  probc[probc>0.999999] = 0.999999
  probc[probc<0.000001] = 0.000001
  like = sum(log(probc))
  return(-like)
}
```

Optimization:

```
optim(par = c(0.5,-0.1), fn=like_fun_2)$par
```

```
## [1]  0.5710928 -0.1401741
```

By optimization results, we have beta equals -0.14. I fail to compute the marginal effect of beta obtained here.

*Exercise*7

In this exercise, we are interested in the effect of excluding choices where the program is "Others".

Q1.Explain and justify, which model (first or second model) you think is appropriate to conduct this exercise:

Answer:

Since we are excluding "Others" program, school quality become less of a program and student test score now has a more direct impact of the three program. Previously, good school and bad school can have different preference over opening programs thus school quality would better reflect that changes. Now, since we are dealing with three major programs, school's quality become less of a concern, and student test score will show more of their preferences over different school-programs. Thus, first model is more appropriate.