A3_1

Peilin Wang

3/22/2022

Exercise 1

```
  #1-1
n_student = nrow(datstu) #340823
n_student
```

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

```
n_school = length(unique(datsss$schoolcode)) #898
n_school
```

```
## [1] 898
```

```
program = c(datstu$choicepgm1,datstu$choicepgm2,datstu$choicepgm3,datstu$choicepgm4,
            datstu$choicepgm5,datstu$choicepgm6)
u_program = unique(program)
u_program = na.omit(u_program)
n_program = length(u_program) #32
n_program
```

```
## [1] 32
```

```
  #1-2
long_dat = datstu %>% gather('schoolcode1':'schoolcode6', key = 'school', value = 'code')
long_dat = long_dat %>% gather('choicepgm1':'choicepgm6', key = 'pgm', value = 'choice')
long_dat$school = substr(long_dat$school, start = 11, stop = 11)
long_dat$pgm = substr(long_dat$pgm, start = 10, stop = 10)
dat_match = long_dat %>% filter(long_dat$school == long_dat$pgm)
nrow(unique(dat_match[c('code','choice')])) #3086
```

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

```
  #1-3
colnames(dat_match)[8] = "schoolcode"
dat_match_1 = merge(dat_match, datsss, by = "schoolcode")
dis_same = dat_match_1 %>% group_by(V1.x) %>% summarise(same = jssdistrict[1] %in% sssdistrict)
table(dis_same$same)["TRUE"] #265464
```

```
##   TRUE
## 265464
```

```
  #1-4
rank = datstu  %>% select(c(2,5:10,17:18))
rank = na.omit(rank)
rank = rank %>% filter(rankplace != 99)
rank$admit_1 = ifelse(rank$rankplace ==1, rank$schoolcode1, 0)
c_1 = count(rank,admit = admit_1)
```

```
rank$admit_2 = ifelse(rank$rankplace == 2, rank$schoolcode2, 0)
c_2 = count(rank,admit = admit_2)
rank$admit_3 = ifelse(rank$rankplace == 3, rank$schoolcode3, 0)
c_3 = count(rank,admit = admit_3)
rank$admit_4 = ifelse(rank$rankplace == 4, rank$schoolcode4, 0)
c_4 = count(rank,admit = admit_4)
rank$admit_5 = ifelse(rank$rankplace == 5, rank$schoolcode5, 0)
c_5 = count(rank,admit = admit_5)
rank$admit_6 = ifelse(rank$rankplace == 6, rank$schoolcode6, 0)
c_6 = count(rank,admit = admit_6)
admit_n = rbind(c_1,c_2,c_3,c_4,c_5,c_6) %>% group_by(admit) %>% summarise(sum(n))
admit_n
```

```
## # A tibble: 518 x 2
##    admit `sum(n)`
##    <dbl>    <int>
##  1     0   658350
##  2 10101      374
##  3 10102      220
##  4 10103      389
##  5 10104      209
##  6 10105      324
##  7 10106      359
##  8 10107      288
##  9 10108      292
## 10 10109      283
## # ... with 508 more rows
```

```
 #1-5
admit = cbind(rank$admit_1, rank$admit_2,rank$admit_3,rank$admit_4,rank$admit_5,rank$admit_6)
Tr_admit = rowSums(admit)
datstu_ad = cbind(rank, Tr_admit)
low_score = datstu_ad %>% group_by(Tr_admit) %>% summarise(min = min(score))
low_score
```

```
## # A tibble: 517 x 2
##    Tr_admit   min
##       <dbl> <dbl>
##  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
## # ... with 507 more rows
```

```
 #1-6
mean_score = datstu_ad %>% group_by(Tr_admit) %>% summarise(mean = mean(score))
mean_score
```

```
## # A tibble: 517 x 2
```

```
##    Tr_admit  mean
##       <dbl> <dbl>
##  1    10101  320.
##  2    10102  394.
##  3    10103  354.
##  4    10104  297.
##  5    10105  351.
##  6    10106  340.
##  7    10107  312.
##  8    10108  303.
##  9    10109  282.
## 10    10110  407.
## # ... with 507 more rows
```

Exercise 2

```
choice1 = datstu %>% select(schoolcode1,choicepgm1)
choice2 = datstu %>% select(schoolcode2,choicepgm2)
choice3 = datstu %>% select(schoolcode3,choicepgm3)
choice4 = datstu %>% select(schoolcode4,choicepgm4)
choice5 = datstu %>% select(schoolcode5,choicepgm5)
choice6 = datstu %>% select(schoolcode6,choicepgm6)
names(choice2) = names(choice1)
names(choice3) = names(choice1)
names(choice4) = names(choice1)
names(choice5) = names(choice1)
names(choice6) = names(choice1)
choices = rbind(choice1,choice2,choice3,choice4,choice5,choice6)
choices = unique(choices)

colnames(choices)[1] = "schoolcode"
colnames(choices)[2] = "choicepgm"
colnames(low_score)[1] = "schoolcode"
colnames(mean_score)[1] = 'schoolcode'
colnames(admit_n)[1] = "schoolcode"
choices = merge(choices,datsss, by = "schoolcode")
choices = merge(choices,low_score, by = "schoolcode")
choices = merge(choices,mean_score, by = "schoolcode")
choices = merge(choices, admit_n, by = "schoolcode")
```

Exercise 3

```
datsss_1 = datsss[!duplicated(datsss$schoolcode),]
colnames(datstu_ad)[15] = "schoolcode"
dist = merge(datstu_ad,datsss_1, by = "schoolcode")
datjss_1 = datjss
colnames(datjss_1)[3] = "jsslong"
colnames(datjss_1)[4] = "jsslat"
dist = merge(dist,datjss_1, by = "jssdistrict")
dist_1 = dist %>% select(ssslong,ssslat,jsslong,jsslat)
dist_1 = dist_1 %>% mutate(dist = sqrt((69.172*(ssslong - jsslong)*cos(jsslat/57.3))^2 + (69.172*(sssla
```

Exercise 4

```
datsu_1 = datstu
datsu_1$scode_rev1 = str_sub(datsu_1$schoolcode1, start = 1, end = 3)
```

```
datsu_1$scode_rev2 = str_sub(datsu_1$schoolcode2, start = 1, end = 3)
datsu_1$scode_rev3 = str_sub(datsu_1$schoolcode3, start = 1, end = 3)
datsu_1$scode_rev4 = str_sub(datsu_1$schoolcode4, start = 1, end = 3)
datsu_1$scode_rev5 = str_sub(datsu_1$schoolcode5, start = 1, end = 3)
datsu_1$scode_rev6 = str_sub(datsu_1$schoolcode6, start = 1, end = 3)

arts = c("General Arts", "Visual Arts")
economics = c("Home Economics", "Business")
science = "General Science"

datsu_1$pgm_rev1 = ifelse(datsu_1$choicepgm1 %in% arts, "arts",
                          ifelse(datsu_1$choicepgm1 %in% economics, "economics",
                                 ifelse(datsu_1$choicepgm1 %in% science, "science",
                                        "others")))
datsu_1$pgm_rev2 = ifelse(datsu_1$choicepgm2 %in% arts, "arts",
                          ifelse(datsu_1$choicepgm2 %in% economics, "economics",
                                 ifelse(datsu_1$choicepgm2 %in% science, "science",
                                        "others")))
datsu_1$pgm_rev3 = ifelse(datsu_1$choicepgm3 %in% arts, "arts",
                          ifelse(datsu_1$choicepgm3 %in% economics, "economics",
                                 ifelse(datsu_1$choicepgm3 %in% science, "science",
                                        "others")))
datsu_1$pgm_rev4 = ifelse(datsu_1$choicepgm4 %in% arts, "arts",
                          ifelse(datsu_1$choicepgm4 %in% economics, "economics",
                                 ifelse(datsu_1$choicepgm4 %in% science, "science",
                                        "others")))
datsu_1$pgm_rev5 = ifelse(datsu_1$choicepgm5 %in% arts, "arts",
                          ifelse(datsu_1$choicepgm5 %in% economics, "economics",
                                 ifelse(datsu_1$choicepgm5 %in% science, "science",
                                        "others")))
datsu_1$pgm_rev6 = ifelse(datsu_1$choicepgm6 %in% arts, "arts",
                          ifelse(datsu_1$choicepgm6 %in% economics, "economics",
                                 ifelse(datsu_1$choicepgm6 %in% science, "science",
                                        "others")))

datsu_1 = datsu_1 %>% mutate(choice_rev1 = paste(scode_rev1,pgm_rev1,sep = ''),
                             choice_rev2 = paste(scode_rev2,pgm_rev2,sep = ''),
                             choice_rev3 = paste(scode_rev3,pgm_rev3,sep = ''),
                             choice_rev4 = paste(scode_rev4,pgm_rev4,sep = ''),
                             choice_rev5 = paste(scode_rev5,pgm_rev5,sep = ''),
                             choice_rev6 = paste(scode_rev6,pgm_rev6,sep = ''))
datsu_1 = datsu_1 %>% filter(rankplace != 99)
datsu_1_ad = function(X){
  X$ad_1 = ifelse(X$rankplace == 1, X$choice_rev1,NA)
  X$ad_2 = ifelse(X$rankplace == 2, X$choice_rev2,NA)
  X$ad_3 = ifelse(X$rankplace == 3, X$choice_rev3,NA)
  X$ad_4 = ifelse(X$rankplace == 4, X$choice_rev4,NA)
  X$ad_5 = ifelse(X$rankplace == 5, X$choice_rev5,NA)
  X$ad_6 = ifelse(X$rankplace == 6, X$choice_rev6,NA)
  A = X %>% as.data.frame(X$ad_1,X$ad_2,X$ad_3,X$ad_4,X$ad_5,X$ad_6)
}
X = datsu_1
C = datsu_1_ad(X)
```

```
C = C %>% unite("ad",ad_1,ad_2,ad_3,ad_4,ad_5,ad_6, na.rm = TRUE, remove = FALSE)
lowscore_rev = C %>% group_by(ad) %>% summarise(minscore = min(score))
cutoff_rev = C %>% group_by(ad) %>% summarise(meanscore = mean(score))
rev = merge(lowscore_rev,cutoff_rev, by = "ad")
C_high = C %>% arrange(desc(score))
C_high = C_high[1:20000,]
```

Exercise 5 #multinomial

```
set.seed(100)
x_1 = sample(1:nrow(C_high),100)
x_2 = C_high[x_1,]
#likelihood function 1st attempt
multi_like_fun = function(param, x_2){
  choice = x_2$choice_rev1
  score = x_2$score
  ni=nrow(x_2)
  nj=length(unique(x_2[,31]))
  ut = mat.or.vec(ni,nj)
  pn1 = param[1:nj]
  pn2 = param[(nj+1):(2*nj)]

  for (j in 1:nj)
  {
    ut[,j] = pn1[j] + score*pn2[j]
  }
  prob   = exp(ut)
  prob   = sweep(prob,MARGIN=1,FUN="/",STATS=rowSums(prob))
  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)
}

  #choice matrix
ni=nrow(x_2)
nj=length(unique(x_2[,31]))
Y=matrix(0, ni,nj)
for(i in 1:nj){
  for(j in 2:ni){
    if(x_2$choice_rev1[j]==i){
      Y[j,i]=1
    }
  }
}
Y[1,1]=1

p=as.matrix(x_2[,2],ncol=1)
#Likelihood Function
```

```r
m_like=function(x,beta) {
  coef=exp(matrix(rep(c(0,beta[1:20]),nrow(x)),byrow=TRUE,nrow(x)) + t(apply(x,1,function(x)x * c(0,beta
  coef_sum=apply(coef,1,sum)
  return(coef/coef_sum)
}
m_llike=function(y,x,beta) {
  lprob=log(m_like(x,beta))
  return(-sum(t(Y) %*% lprob))
}
#optimization
model_1=optim(function(beta) m_llike(y=y,x=p,b=beta),par=runif(40),method="BFGS")
as.matrix(model_1$par)
```

```
##               [,1]
##  [1,]  1.229075481
##  [2,]  0.768295350
##  [3,]  0.785073360
##  [4,]  0.903431930
##  [5,]  1.085116373
##  [6,]  1.163825219
##  [7,]  0.682981942
##  [8,]  1.332842683
##  [9,]  1.077453483
## [10,]  0.793212027
## [11,]  0.937043597
## [12,]  0.667924621
## [13,]  1.017534482
## [14,]  1.291674695
## [15,]  1.021772134
## [16,]  0.551192188
## [17,]  0.723570075
## [18,]  0.497620308
## [19,]  0.413173985
## [20,]  0.904314144
## [21,] -0.003602831
## [22,] -0.002330360
## [23,] -0.002412664
## [24,] -0.002710409
## [25,] -0.003218765
## [26,] -0.003432549
## [27,] -0.002104091
## [28,] -0.003902355
## [29,] -0.003177278
## [30,] -0.002413699
## [31,] -0.002741576
## [32,] -0.002084813
## [33,] -0.003053964
## [34,] -0.003777296
## [35,] -0.003041322
## [36,] -0.001726680
## [37,] -0.002205447
## [38,] -0.001572963
## [39,] -0.001316723
## [40,] -0.002695714
```

```
#marginal effect
pij_m1=m_like(p,model_1$par)
mb=c(0,model_1$par[21:40])
me_model1=array(0,dim=c(nrow(p),21))
for (i in 1:nrow(p)) {
  be=sum(pij_m1[i,]*mb)
  for (j in 1:21) {
    me_model1[i,j] <- pij_m1[i,j]*(mb[j]-be)
  }
}
for (i in 1:nrow(p)) {
  be=sum(pij_m1[i,]*mb)
  me_model1[i,]=pij_m1[i,]*(mb-be)
}
me_model1=apply(me_model1, 2, mean)
me_model1
```

```
##  [1]  1.365950e-04 -5.023586e-05  9.002269e-06  5.003613e-06 -9.017196e-06
##  [6] -3.254450e-05 -4.234671e-05  1.982257e-05 -6.340855e-05 -3.086867e-05
## [11]  4.993486e-06 -1.071628e-05  2.058517e-05 -2.477131e-05 -5.807014e-05
## [16] -2.440664e-05  3.835410e-05  1.500669e-05  4.604879e-05  5.935293e-05
## [21] -8.378779e-06
```

Exercise 6 #conditional logit

```
set.seed(100)
names(cutoff_rev)[1] = "choice_rev1"
C_high_1 = merge(x_2,cutoff_rev, by = "choice_rev1" )

q_school = matrix(as.numeric(C_high_1[,44],ncol=1))
#likelihood function 1st attempt
ni_2 = nrow(C_high_1)
nj_2 = length(unique(C_high_1[,44]))
Y_2 = matrix(0, ni_2,nj_2)
for(i in 1:nj_2){
  for(j in 2:ni_2){
    if(C_high_1$choice_rev1[j]==i){
      Y_2[j,i]=1
    }
  }
}
Y_2[1,1]=1

likelihood=function(x,beta) {
  coef=exp(as.matrix(rep(1, nrow(x))) %*% c(0,beta[1:39]) + t(x) %*% beta[40])
  coef_sum=apply(coef,1,sum)
  return(coef/coef_sum)
}
llike=function(y1,x,beta) {
  lprob=log(likelihood(x,beta))
  return(-sum(t(Y_2) %*% lprob))
}
model1=optim((function(beta) llike(y,q_school,beta)),par=runif(40),method="BFGS")
as.matrix(model1$par)
```

```r
#marginal effect
pij=likelihood(q_school,model1$par)
mid=array(0,dim = c(nrow(q_school),40,40))
for (i in 1:nrow(q_school)) {
  diag(mid[i,,]) <- 1
}
llikem=array(0,dim=c(nrow(q_school),40,40))
for (i in 1:nrow(q_school)) {
  for (j in 1:40) {
    for (k in 1:40) {
      llikem[i,j,k]=pij[i,j]*(mid[i,j,k]-pij[i,k])*model1$par[40]
    }
  }
}
me_model1=apply(llikem,c(2,3),mean)

#likelihood function 2nd attempt
con_like_fun = function(param,C_high_1){
  school_q = C_high_1$meanscore
  ch = C_high_1$choice_rev1
  ni = nrow(C_high_1)
  nj=length(unique(C_high_1[,44]))
  ut = mat.or.vec(ni,nj)
  for (j in 2:nj)
  {
    ut[,j] = param[1] + param[2] * school_q[j]
  }
  prob  = exp(ut)
  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))
  return(-like)
}
start = runif(80,0,1)
res  = optim(start,fn=con_like_fun,method="BFGS",control=list(trace=6,REPORT=1,maxit=3000),C_high_1 = C
res$par
```

Exercise 7

I think the second model is more appropriate to conduct the exercise. The first model is the multinomial logit model which uses scores as invariant regressors. It changes relying on the changes of student. When excluding others in program, there are fewer choices remaining, which means the school quality changes. The second model is the conditional logit model. It has the regressor school quality. So, the second model can reflect the appropriate change of school quality.

```r
C_high_2 = x_2 %>% filter(pgm_rev1 != "others") %>% filter(pgm_rev2 != "others") %>%
                   filter(pgm_rev3 != "others") %>% filter(pgm_rev4 != "others") %>%
                   filter(pgm_rev5 != "others") %>% filter(pgm_rev6 != "others")
```