library(readr)

library(tidyverse)

library(data.table)

library(ggplot2)

library(fastDummies)

library(nnet)

# Import Datasets

datjss <- read\_csv("A3/Data/datjss.csv")

datsss <- read\_csv("A3/Data/datsss.csv")

datstu <- read\_csv("A3/Data/datstu\_v2.csv")

datstu <- subset(datstu,score!='NA'&rankplace!='NA')

datsss <- subset(datsss,schoolname!='NA')

datsss <- datsss %>% distinct(schoolcode,.keep\_all = TRUE)

#Exercise 1

Nstudent <- nrow(datstu)

Nschool <- n\_distinct(datsss$schoolcode)

programs <- c(datstu$choicepgm1,datstu$choicepgm2,datstu$choicepgm3,datstu$choicepgm4,datstu$choicepgm5,datstu$choicepgm6)

programs <- na.omit(programs)

Nprogram <- n\_distinct(programs)

Ldatstu <- datstu %>% gather('schoolcode1':'schoolcode6', key = 'school', value = 'schoice')

Ldatstu <- Ldatstu %>% gather('choicepgm1':'choicepgm6', key = 'pgm', value = 'pchoice')

Ldatstu$school <- str\_sub(Ldatstu$school, start = 11, end = 11)

Ldatstu$school <- as.numeric(Ldatstu$school)

Ldatstu$pgm <- str\_sub(Ldatstu$pgm, start = 10, end = 10)

Ldatstu$pgm <- as.numeric(Ldatstu$pgm)

Ldatstu\_matching <- subset(Ldatstu,Ldatstu$school==Ldatstu$pgm&Ldatstu$schoice!='NA'&Ldatstu$pchoice!='NA')

Nchoice <- nrow(unique(Ldatstu\_matching[c('schoice','pchoice')]))

Ldatstu\_matching\_sss <- Ldatstu\_matching %>% rename(schoolcode=schoice)

Ldatstu\_matching\_sss <- merge(Ldatstu\_matching\_sss,datsss,by="schoolcode")

datasame <- Ldatstu\_matching\_sss %>% group\_by(V1.x) %>% summarise(same = jssdistrict[1] %in% sssdistrict)

datasame$same <- as.numeric(datasame$same)

Nsame <- sum(datasame$same)

Ldatstu\_matching\_sss\_admit <- subset(Ldatstu\_matching\_sss,rankplace!=99)

Ldatstu\_matching\_sss\_admit <- subset(Ldatstu\_matching\_sss\_admit,rankplace==school)

Nadmit <- data.frame(table(Ldatstu\_matching\_sss\_admit$schoolcode))

Nadmit <- Nadmit %>% rename(schoolcode=Var1)

Nadmit <- Nadmit %>% rename(Nadmit=Freq)

lowscore <- Ldatstu\_matching\_sss\_admit %>% group\_by(schoolcode) %>% summarise(minscore=min(score))

quality <- Ldatstu\_matching\_sss\_admit %>% group\_by(schoolcode) %>% summarise(meanscore=mean(score))

#Exercise 2

Ldatstu\_matching\_sss\_admit$choice <- paste(Ldatstu\_matching\_sss\_admit$schoolcode,Ldatstu\_matching\_sss\_admit$pchoice,sep = ' ')

Ncadmit <- data.frame(table(Ldatstu\_matching\_sss\_admit$choice))

Ncadmit <- Ncadmit %>% rename(choice=Var1)

Ncadmit <- Ncadmit %>% rename(people=Freq)

clowscore <- Ldatstu\_matching\_sss\_admit %>% group\_by(choice) %>% summarise(minscore=min(score))

cquality <- Ldatstu\_matching\_sss\_admit %>% group\_by(choice) %>% summarise(meanscore=mean(score))

schooldata <- unique(Ldatstu\_matching\_sss\_admit[c('schoolcode','pchoice')])

schooldata$choice <- paste(schooldata$schoolcode,schooldata$pchoice,sep = ' ')

schooldata <- merge(datsss,schooldata,by='schoolcode')

schooldata <- merge(schooldata,Ncadmit,by='choice')

schooldata <- merge(schooldata,clowscore,by='choice')

schooldata <- merge(schooldata,cquality,by='choice')

schooldataf <-data.frame(unique(Ldatstu\_matching[c('schoice','pchoice')]))

schooldataf$choice <- paste(schooldataf$schoice,schooldataf$pchoice,sep = ' ')

schooldataf <- schooldataf %>% rename(schoolcode=schoice)

schooldataf <- merge(schooldataf,datsss,by='schoolcode')

schooldata1 <- left\_join(schooldataf,schooldata,by='choice')

schooldata <- subset(schooldata1,select = -c(9:15))

schooldata <- schooldata[is.na(schooldata)] <- 0

#Exercise 3

Ldatstu\_matching\_sss\_jss <- merge(Ldatstu\_matching\_sss,datjss,by='jssdistrict')

Ldatstu\_matching\_sss\_jss <- Ldatstu\_matching\_sss\_jss %>% rename(jsslong = point\_x)

Ldatstu\_matching\_sss\_jss <- Ldatstu\_matching\_sss\_jss %>% rename(jsslat = point\_y)

attach(Ldatstu\_matching\_sss\_jss)

Ldatstu\_matching\_sss\_jss$distance <- sqrt((69.172 \* (ssslong-jsslong) \* cos(jsslat/57.3))^2 + (69.172 \* (ssslat-jsslat))^2)

detach(Ldatstu\_matching\_sss\_jss)

#Exercise 4

Ldatstu\_matching\_sss\_jss$scode\_rev <- str\_sub(Ldatstu\_matching\_sss\_jss$schoolcode, start = 1, end = 3)

Ldatstu\_matching\_sss\_jss$pchoice <- as\_factor(Ldatstu\_matching\_sss\_jss$pchoice)

Ldatstu\_matching\_sss\_jss$pgm\_rev <- Ldatstu\_matching\_sss\_jss$pchoice %>% fct\_collapse(arts = c("General Arts","Visual Arts"),economics = c("Business","Home Economics"),science="General Science",other\_level = 'others')

levels(Ldatstu\_matching\_sss\_jss$pgm\_rev)

Ldatstu\_matching\_sss\_jss$choice\_rev <- paste(Ldatstu\_matching\_sss\_jss$scode\_rev,Ldatstu\_matching\_sss\_jss$pgm\_rev,sep = ' ')

lowscore\_rev <- Ldatstu\_matching\_sss\_jss %>% group\_by(choice\_rev) %>% summarise(minscore=min(score))

quality\_rev <- Ldatstu\_matching\_sss\_jss %>% group\_by(choice\_rev) %>% summarise(meanscore=mean(score))

#Exercise 5

Ldatstu\_matching\_sss\_jss\_first <- subset(Ldatstu\_matching\_sss\_jss,school==1)

Ldatstu\_matching\_sss\_jss\_first$choice\_rev <- as.factor(Ldatstu\_matching\_sss\_jss\_first$choice\_rev)

Ldatstu\_matching\_sss\_jss\_first$choice\_rev <- as.numeric(Ldatstu\_matching\_sss\_jss\_first$choice\_rev)

like\_fun = function(param,data)

{

score = data$score

ch = data$choice\_rev

ni = nrow(data)

nj = length(unique(data[,22]))

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] = pn1[j] + score\*pn2[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)

}

x <-sample(1:nrow(Ldatstu\_matching\_sss\_jss\_first),100)

sample <- Ldatstu\_matching\_sss\_jss\_first[x,]

m <-length(unique(sample[,22]))

sample$choice\_rev<-as.factor(sample$choice\_rev)

sample$choice\_rev<-as.numeric(sample$choice\_rev)

start1 = runif(130,-1,1)

res = optim(start,fn=like\_fun,method="BFGS",control=list(trace=6,REPORT=1,maxit=10000),data=sample,hessian=TRUE)

mymodel = multinom(choice\_rev~score,data=Ldatstu\_matching\_sss\_jss\_first)

reg = summary(mymodel)

start1 = runif(422,0,1)

res = optim(start,fn=like\_fun,method="BFGS",control=list(trace=6,REPORT=1,maxit=10000),data=sample,hessian=TRUE)

#Exercise 6

#Exercise 7