Assignment2

Luyi Huang

2021/2/25

Exercise 1 Data Desription

```
library(bayesm)
```

```
## Warning: package 'bayesm' was built under R version 4.0.4
```

```
data(margarine)
choiceprice=margarine$choicePrice
demos=margarine$demos
all=merge(choiceprice,demos,by="hhid")
```

```
avg=apply(as.matrix(choiceprice[,3:12]),2, mean)
print(avg)
```

```
##   PPk_Stk   PBB_Stk   PFl_Stk  PHse_Stk  PGen_Stk  PImp_Stk   PSS_Tub   PPk_Tub
## 0.5184362 0.5432103 1.0150201 0.4371477 0.3452819 0.7807785 0.8250895 1.0774094
##   PFl_Tub  PHse_Tub
## 1.1893758 0.5686734
```

```
std<-function(x) sd(x)/sqrt(length(x))
stdc=apply(as.matrix(choiceprice[,3:12]),2, std)
print(stdc)
```

```
##      PPk_Stk      PBB_Stk      PFl_Stk     PHse_Stk     PGen_Stk     PImp_Stk
## 0.0022512977 0.0017998109 0.0006415859 0.0017773659 0.0005259808 0.0017147682
##      PSS_Tub      PPk_Tub      PFl_Tub     PHse_Tub
## 0.0009155455 0.0004446156 0.0002102142 0.0010837139
```

```
#Average and dispersion in product characteristics
library(tidyr)
library(dplyr)
#Compute all summary statistics including both chosen products and non-chosen products
means <- t(all %>% summarise_at(3:12,mean))
mins <- t(all %>% summarise_at(3:12,min))
maxs <- t(all %>% summarise_at(3:12,max))
sds <- t(all %>% summarise_at(3:12,sd))
vars <- t(all %>% summarise_at(3:12,var))
```

```r
des1=cbind(means,mins,maxs,sds,vars)
label1=c("mean","min","max","sd","var")
colnames(des1) <- label1
des1=round(des1,digits=3)
des1
```

```
##           mean  min  max    sd   var
## PPk_Stk  0.518 0.19 0.67 0.151 0.023
## PBB_Stk  0.543 0.19 1.01 0.120 0.014
## PFl_Stk  1.015 0.95 1.16 0.043 0.002
## PHse_Stk 0.437 0.19 0.64 0.119 0.014
## PGen_Stk 0.345 0.25 0.55 0.035 0.001
## PImp_Stk 0.781 0.33 2.30 0.115 0.013
## PSS_Tub  0.825 0.50 0.98 0.061 0.004
## PPk_Tub  1.077 0.98 1.24 0.030 0.001
## PFl_Tub  1.189 0.69 1.47 0.014 0.000
## PHse_Tub 0.569 0.33 1.27 0.072 0.005
```

Market Share :

```r
ms<-as.data.frame(choiceprice %>% count(choice))
ms$n<-ms$n/nrow(all)
ms
```

```
##    choice          n
## 1       1 0.39507830
## 2       2 0.15637584
## 3       3 0.05436242
## 4       4 0.13266219
## 5       5 0.07046980
## 6       6 0.01655481
## 7       7 0.07136465
## 8       8 0.04541387
## 9       9 0.05033557
## 10     10 0.00738255
```

Market Share by Characteristics:

```r
#finding the price
for (i in 1:nrow(all)){
  all$choiceprice[i]=all[i,all$choice[i]+2]
  all$choices[i]=colnames(all)[all$choice[i]+2]
}
```

```r
means <- as.data.frame((t(all %>% summarise_at(3:12,mean))))
means <- cbind(choices = rownames(means), means)
rownames(means) <- 1:nrow(means)
means
```

```
##     choices        V1
## 1   PPk_Stk 0.5184362
```

```
## 2   PBB_Stk 0.5432103
## 3   PFl_Stk 1.0150201
## 4  PHse_Stk 0.4371477
## 5  PGen_Stk 0.3452819
## 6  PImp_Stk 0.7807785
## 7   PSS_Tub 0.8250895
## 8   PPk_Tub 1.0774094
## 9   PFl_Tub 1.1893758
## 10 PHse_Tub 0.5686734
```

```
all=left_join(all, unique(means), by = c("choices"="choices"))
```

```
all$indict<-all$choiceprice>all$V1
sum(all$indict)
```

```
## [1] 2091
```

```
all%>%
  filter(all$indict==TRUE) %>%
  count(choice)%>%
  mutate(n=n/sum(all$indict))
```

```
##    choice           n
## 1       1 0.376853180
## 2       2 0.125777140
## 3       3 0.025346724
## 4       4 0.141559063
## 5       5 0.066953611
## 6       6 0.008608321
## 7       7 0.095648015
## 8       8 0.055475849
## 9       9 0.095648015
## 10     10 0.008130081
```

```
all%>%
  filter(all$indict==TRUE) %>%
  count(choice)%>%
  mutate(n=n/(nrow(all)-sum(all$indict)))
```

```
##    choice           n
## 1       1 0.331231610
## 2       2 0.110550652
## 3       3 0.022278268
## 4       4 0.124422026
## 5       5 0.058848256
## 6       6 0.007566204
## 7       7 0.084068937
## 8       8 0.048759983
## 9       9 0.084068937
## 10     10 0.007145860
```

consider mapping on choices and attributes

```r
des3= all %>%
  group_by(choice) %>%
  summarize(
    famsize1_2=sum(Fs3_4 == 0 & Fs5.==0),
    famsize3_4=sum(Fs3_4 == 1 & Fs5.==0),
    famsize5.=sum(Fs3_4 == 0 & Fs5.==1),
    college=sum(college==1),
    whtcollar=sum(whtcollar==1),
    retired=sum(retired==1)
  )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
notdes3= all %>%
  group_by(choice) %>%
  summarize(
    notcollege=sum(college==0),
    notwhtcollar=sum(whtcollar==0),
    notretired=sum(retired==0)
  )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
des3=merge(des3,notdes3)
des3
```

```
##    choice famsize1_2 famsize3_4 famsize5. college whtcollar retired notcollege
## 1       1        622        902       242     561      1007     352       1205
## 2       2        261        360        78     219       380     168        480
## 3       3        161         62        20     110       132     129        133
## 4       4        177        298       118     174       351      91        419
## 5       5         65        187        63      86       225      46        229
## 6       6         33         18        23      32        42      28         42
## 7       7        142        157        20     103       184      47        216
## 8       8         70        122        11      52       116      20        151
## 9       9        146         68        11      62       130      81        163
## 10     10          3         12        18      15        31       4         18
##    notwhtcollar notretired
## 1           759       1414
## 2           319        531
## 3           111        114
## 4           242        502
## 5            90        269
## 6            32         46
## 7           135        272
## 8            87        183
## 9            95        144
## 10            2         29
```

Larger familisize, more retired people will like choice1. Overall, choice1 is much more popular among all the attributes of customers.

Exercise 2 First Model

consider using conditional logit model since the price value varies of the same alternate accross people. Likelihood and Optimization:

```r
library(mlogit)
```

```
## Warning: package 'mlogit' was built under R version 4.0.4

## Loading required package: dfidx

## Warning: package 'dfidx' was built under R version 4.0.4

##
## Attaching package: 'dfidx'

## The following object is masked from 'package:stats':
##
##     filter
```

```r
library(stargazer)
```

```
## Warning: package 'stargazer' was built under R version 4.0.3

##
## Please cite as:

##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```r
library(texreg)
```

```
## Warning: package 'texreg' was built under R version 4.0.3

## Version:  1.37.5
## Date:     2020-06-17
## Author:   Philip Leifeld (University of Essex)
##
## Consider submitting praise using the praise or praise_interactive functions.
## Please cite the JSS article in your publications -- see citation("texreg").

##
## Attaching package: 'texreg'

## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
library(survival)
library(nnet)
price=margarine$choicePrice
colnames(price)[3:12]=paste0("price",1:10)
price<-price[-c(1)]
clogit0=mlogit.data(price,varying=2:11,shape="wide",sep="",choice="choice")
clogit1=mlogit(choice~price,data=clogit0)
summary(clogit1)
```

```
##
## Call:
## mlogit(formula = choice ~ price, data = clogit0, method = "nr")
##
## Frequencies of alternatives:choice
##         1         2         3         4         5         6         7         8
## 0.3950783 0.1563758 0.0543624 0.1326622 0.0704698 0.0165548 0.0713647 0.0454139
##         9        10
## 0.0503356 0.0073826
##
## nr method
## 6 iterations, 0h:0m:2s
## g'(-H)^-1g = 2.19E-08
## gradient close to zero
##
## Coefficients :
##                 Estimate Std. Error  z-value  Pr(>|z|)
## (Intercept):2  -0.954307   0.050046 -19.0685 < 2.2e-16 ***
## (Intercept):3   1.296968   0.108651  11.9370 < 2.2e-16 ***
## (Intercept):4  -1.717332   0.054158 -31.7096 < 2.2e-16 ***
## (Intercept):5  -2.904005   0.071461 -40.6379 < 2.2e-16 ***
## (Intercept):6  -1.515311   0.126230 -12.0043 < 2.2e-16 ***
## (Intercept):7   0.251768   0.079164   3.1803  0.001471 **
## (Intercept):8   1.464868   0.118047  12.4092 < 2.2e-16 ***
## (Intercept):9   2.357505   0.133774  17.6230 < 2.2e-16 ***
## (Intercept):10 -3.896593   0.177419 -21.9627 < 2.2e-16 ***
## price          -6.656580   0.174279 -38.1949 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -7464.9
## McFadden R^2:  0.099075
## Likelihood ratio test : chisq = 1641.8 (p.value = < 2.22e-16)
```

```r
ni=nrow(price)
nj=ncol(price[,2:11])
Y=class.ind(all$choice)
#consider likelihood, we only have one variable price and thus beta is a scalor in the function.
clogit_ll<-function(beta){

  #Create the constant as instructed
  intercept=cbind(0,matrix(rep(beta[1:nj-1],each=ni),ni,nj-1))

  #Use the lecture definition of conditional logit to compute the likelihood
```

```r
  XB=price[,2:11]*beta[nj]
  XB=intercept+XB
  eXB=exp(XB)
  teXB=rowSums(eXB)
  prob=eXB/teXB

  #Compute the neg log likelihood for each choice using the choice matrix
  llik=sum(Y*log(prob))
  return(-llik)
}
```

```r
set.seed(0)
clogit <- optim(runif(10,-0.1,0.1),clogit_ll,method="BFGS")
```

We can see that the fixed effects of choices varies and increasing prices will reduce the probabiliy of purchasing.

```r
clogit$par
```

```
##  [1] -0.9543264  1.2969599 -1.7173741 -2.9040330 -1.5153099  0.2516940
##  [7]  1.4647896  2.3573682 -3.8966223 -6.6565265
```

Interpretation: Overall, the price has a negative impact on the probability of getting chosen by customers. ## Exercise 3 Second Model The second model should be the multinomial logit model since income varies across different household id.

```r
colnames(all)[3:12]=paste0("price",1:10)
mlogit0=mlogit.data(all,varying=3:12,shape="wide",sep="",choice="choice")
mlogit1=mlogit(choice~0 | Income,data=mlogit0)
summary(mlogit1)
```

```
##
## Call:
## mlogit(formula = choice ~ 0 | Income, data = mlogit0, method = "nr")
##
## Frequencies of alternatives:choice
##         1         2         3         4         5         6         7         8
## 0.3950783 0.1563758 0.0543624 0.1326622 0.0704698 0.0165548 0.0713647 0.0454139
##         9        10
## 0.0503356 0.0073826
##
## nr method
## 6 iterations, 0h:0m:2s
## g'(-H)^-1g = 0.000261
## successive function values within tolerance limits
##
## Coefficients :
##                 Estimate Std. Error  z-value  Pr(>|z|)
## (Intercept):2  -0.8453241  0.0931354  -9.0763 < 2.2e-16 ***
## (Intercept):3  -2.3998575  0.1335802 -17.9657 < 2.2e-16 ***
## (Intercept):4  -1.2013265  0.0971021 -12.3718 < 2.2e-16 ***
## (Intercept):5  -1.6905817  0.1269952 -13.3122 < 2.2e-16 ***
```

```
## (Intercept):6  -4.1397653  0.2109890 -19.6208 < 2.2e-16 ***
## (Intercept):7  -1.5310415  0.1280434 -11.9572 < 2.2e-16 ***
## (Intercept):8  -2.8483522  0.1393848 -20.4352 < 2.2e-16 ***
## (Intercept):9  -2.5755972  0.1361400 -18.9187 < 2.2e-16 ***
## (Intercept):10 -4.2822699  0.3457920 -12.3839 < 2.2e-16 ***
## Income:2       -0.0030887  0.0031140  -0.9919 0.3212477
## Income:3        0.0145862  0.0038255   3.8129 0.0001373 ***
## Income:4        0.0040504  0.0030926   1.3097 0.1902878
## Income:5       -0.0012536  0.0042024  -0.2983 0.7654694
## Income:6        0.0306120  0.0046740   6.5494 5.775e-11 ***
## Income:7       -0.0069326  0.0044161  -1.5698 0.1164518
## Income:8        0.0228862  0.0036217   6.3192 2.629e-10 ***
## Income:9        0.0177430  0.0037623   4.7160 2.405e-06 ***
## Income:10       0.0107909  0.0101300   1.0652 0.2867676
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -8236.8
## McFadden R^2:  0.0059257
## Likelihood ratio test : chisq = 98.199 (p.value = < 2.22e-16)
```

```r
mlogit_ll<-function(beta){

  #Create the constant as instructed
  intercept1=cbind(0,matrix(rep(beta[1:9],each=ni),ni,nj-1))
  intercept2=cbind(0,matrix(rep(beta[10:18],each=ni),ni,nj-1))
  #Use the lecture definition of conditional logit to compute the likelihood
  XB=intercept1+intercept2*cbind(all[,13],all[,13],all[,13],all[,13],all[,13],all[,13],all[,13],all[,13
  eXB=exp(XB)
  teXB=rowSums(eXB)
  prob=eXB/teXB

  #Compute the neg log likelihood for each choice using the choice matrix
  ll=-sum(Y*log(prob))
  return(ll)
}
```

```r
set.seed(0)
model2 <- optim(runif(18,-0.1,0.1),mlogit_ll,method="BFGS")
model2$par[10:18]
```

```
## [1] -0.003156534  0.014502403  0.003978203 -0.001326591  0.030522991
## [6] -0.007003193  0.022804588  0.017661189  0.010711328
```

Income will have a negative impact of getting hired for choice2, choice7, while other choices have the attributes that if the income is higher, the probability of choosing will get higher. ## Exercise 4 Marginal Effects of First and Second Model

```r
beta=matrix(0,nrow=1,ncol=11)
beta[1,1]=0
beta[1,2:11]=clogit$par
delta<-diag(1,10,10)
```

```r
mf=matrix(0,nrow=10,ncol=10)
for (i in 1:nrow(all)){
  intercept=beta[1:10]

  #Use the lecture definition of conditional logit to compute the likelihood
  xibi=as.matrix(exp(price[i,2:11]*beta[11]+intercept))
  xib=rowSums(xibi)
  pij=xibi/xib
 dpij<-matrix(NA,10,10)
  delta<-diag(1,10,10)
 for (k in 1:10){
    delta[,k]=delta[,k]-pij
 }
 for (m in 1:10){
    dpij[m,]=pij*delta[m,]*beta[11]
 }
 mf=mf+dpij
}
```

```r
mf/4470
```

```
##              [,1]         [,2]         [,3]         [,4]         [,5]
##  [1,] -1.28527146  0.295376271  0.120717982  0.295081295  0.156227643
##  [2,]  0.29537627 -0.745425018  0.055081688  0.133449327  0.072823337
##  [3,]  0.12071798  0.055081688 -0.337462134  0.050544479  0.030281618
##  [4,]  0.29508129  0.133449327  0.050544479 -0.712646397  0.064013220
##  [5,]  0.15622764  0.072823337  0.030281618  0.064013220 -0.428073176
##  [6,]  0.03732247  0.016726466  0.007105131  0.016551128  0.008748786
##  [7,]  0.15359412  0.069268799  0.029268706  0.063740223  0.037946081
##  [8,]  0.09929391  0.045205301  0.019664695  0.039259911  0.025088824
##  [9,]  0.11081419  0.050695663  0.021753317  0.044149386  0.028517028
## [10,]  0.01684357  0.006798165  0.003044518  0.005857429  0.004426637
##                [,6]         [,7]         [,8]         [,9]         [,10]
##  [1,]  0.0373224742  0.153594121  0.099293911  0.110814188  0.0168435746
##  [2,]  0.0167264660  0.069268799  0.045205301  0.050695663  0.0067981654
##  [3,]  0.0071051313  0.029268706  0.019664695  0.021753317  0.0030445175
##  [4,]  0.0165511282  0.063740223  0.039259911  0.044149386  0.0058574290
##  [5,]  0.0087487861  0.037946081  0.025088824  0.028517028  0.0044266374
##  [6,] -0.1073254358  0.008537803  0.005430211  0.006113285  0.0007901501
##  [7,]  0.0085378034 -0.420279477  0.025791473  0.027918472  0.0042137976
##  [8,]  0.0054302111  0.025791473 -0.282454942  0.019787183  0.0029334322
##  [9,]  0.0061132854  0.027918472  0.019787183 -0.313030336  0.0032818145
## [10,]  0.0007901501  0.004213798  0.002933432  0.003281814 -0.0481895182
```

```r
z<-with(clogit0,data.frame(price=tapply(price,idx(clogit0,2),mean)))
effects(clogit1,covariate="price",data=z)
```

```
##             1            2            3            4            5
## 1  -1.62005803  0.380917747  0.156525493  0.359808635  0.202433528
## 2   0.38091060 -0.785526022  0.051110136  0.117488008  0.066100448
## 3   0.15652172  0.051109862 -0.352892054  0.048277535  0.027161637
## 4   0.35980170  0.117487949  0.048277769 -0.748505344  0.062437364
```

```
## 5   0.20242887  0.066100166  0.027161667  0.062437129 -0.448427420
## 6   0.04470836  0.014598856  0.005998915  0.013789839  0.007758362
## 7   0.19486054  0.063628840  0.026146158  0.060102755  0.033814677
## 8   0.12221659  0.039908027  0.016398878  0.037696466  0.021208575
## 9   0.14161589  0.046242583  0.019001853  0.043679983  0.024574988
## 10  0.01695855  0.005537566  0.002275479  0.005230694  0.002942864
##                 6            7            8            9           10
## 1   0.0447095545  0.194865062  0.122219636  0.141619353  0.0169590186
## 2   0.0145989729  0.063629123  0.039908274  0.046242847  0.0055376140
## 3   0.0059989307  0.026146134  0.016398891  0.019001860  0.0022754863
## 4   0.0137899426  0.060102992  0.037696680  0.043680210  0.0052307364
## 5   0.0077583914  0.033814683  0.021208616  0.024575024  0.0029428767
## 6  -0.1050843433  0.007468297  0.004684126  0.005427630  0.0006499625
## 7   0.0074683239 -0.432926043  0.020415677  0.023656223  0.0028328496
## 8   0.0046841349  0.020415642 -0.279142265  0.014837190  0.0017767641
## 9   0.0054276423  0.023656193  0.014837197 -0.321095113  0.0020587878
## 10  0.0006499621  0.002832838  0.001776760  0.002058782 -0.0402634956
```

We can see that diagonal elements are almost negative, meaning that price will reduce the chance of getting chosen. The off-diagonal elements are almost positive, meaning that these products are subsititutes.

```
beta=matrix(0,nrow=1,ncol=20)
beta[1,1]=0
beta[1,2:10]=model2$par[1:9]
beta[1,11]=0
beta[1,12:20]=model2$par[10:18]
```

Marginal Effects of MNL:

```
intercept1=beta[1:10]
intercept2=beta[11:20]
mf=matrix(0,nrow=1,ncol=10)
for (i in 1:nrow(all)){
  xibi=exp(intercept1+intercept2*cbind(all[i,13],all[i,13],all[i,13],all[i,13],all[i,13],all[i,13],all[i
  xib=rowSums(xibi)
  pij=xibi/xib
  betai=rowSums(pij*intercept2)
  dpixi=pij*(intercept2-betai)
  mf=mf+dpixi
}
```

check with packages

```
z<-with(mlogit0,data.frame(Income=tapply(Income,idx(clogit0,2),mean)))
effects(mlogit1,covariate="Income",data=z)
```

```
##             1             2             3             4             5
## -1.062467e-03 -9.035571e-04  6.442588e-04  1.849436e-04 -2.781303e-04
##             6             7             8             9            10
##  4.131162e-04 -6.824169e-04  8.781123e-04  7.459655e-04  6.017529e-05
```

```r
mf/4470
```

```
##              [,1]          [,2]         [,3]         [,4]          [,5]
## [1,] -0.001050102 -0.0009015754 0.0006264986 0.0001658639 -0.0002793115
##              [,6]          [,7]         [,8]        [,9]        [,10]
## [1,] 0.0004430249 -0.0006821461 0.0008860677 0.000733821 5.785854e-05
```

As income goes up, choice1, chocie2, choice5, choice7 will have lower probabiltiy of getting purchased. But overall, these coeffcients are small.

Exercise 5 Mixed Logit

```r
mixed_ll<-function(para){
beta1=matrix(0,nrow=1,ncol=11)
gamma1=matrix(0,nrow=1,ncol=20)
beta1[1,1]=0
beta1[1,2:10]=para[1:9]
beta1[1,11]=para[10]
gamma1[1,1]=0
gamma1[1,2:10]=para[1:9]
gamma1[1,11]=0
gamma1[1,12:20]=para[11:19]
intercept=cbind(0,matrix(rep(beta1[2:10],each=ni),ni,nj-1))/2
intercept1=cbind(0,matrix(rep(gamma1[2:10],each=ni),ni,nj-1))/2
intercept2=cbind(0,matrix(rep(gamma1[12:20],each=ni),ni,nj-1))
XB=intercept1+intercept2*cbind(all[,13],all[,13],all[,13],all[,13],all[,13],all[,13],all[,13],all[,13],a
WG=price[,2:11]*beta1[11]+intercept
e=exp(XB+WG)
se=rowSums(e)
 prob=e/se

  #Compute the neg log likelihood for each choice using the choice matrix
  ll=sum(Y*log(prob))
  return(-ll)
}
```

```r
set.seed(0)
model3 <-  optim(runif(19,-0.1,0.1),mixed_ll,method="BFGS", hessian = FALSE)
```

```r
model3$par
```

```
##  [1] -0.838765940  0.891349297 -1.826505215 -2.871274491 -2.454313591
##  [6]  0.498899430  0.805743197  1.866687349 -4.139798727 -6.659769251
## [11] -0.004332423  0.014249506  0.004024862 -0.001264060  0.029718813
## [16] -0.009321103  0.021905634  0.016902603  0.008665919
```

consider alternative specification:

```r
price[,12]=all$Income
colnames(price)[12] <- "Income"
logit0=mlogit.data(price,varying=2:11,shape="wide",sep="",choice="choice")
logit1=mlogit(choice~price|Income,data=logit0)
summary(logit1)
```

```
##
## Call:
## mlogit(formula = choice ~ price | Income, data = logit0, method = "nr")
##
## Frequencies of alternatives:choice
##         1         2         3         4         5         6         7         8
## 0.3950783 0.1563758 0.0543624 0.1326622 0.0704698 0.0165548 0.0713647 0.0454139
##         9        10
## 0.0503356 0.0073826
##
## nr method
## 6 iterations, 0h:0m:2s
## g'(-H)^-1g = 4.23E-08
## gradient close to zero
##
## Coefficients :
##                  Estimate Std. Error  z-value  Pr(>|z|)
## (Intercept):2  -0.8406734  0.1038446  -8.0955 6.661e-16 ***
## (Intercept):3   0.8886069  0.1594585   5.5727 2.509e-08 ***
## (Intercept):4  -1.8284916  0.1032180 -17.7149 < 2.2e-16 ***
## (Intercept):5  -2.8734106  0.1347573 -21.3229 < 2.2e-16 ***
## (Intercept):6  -2.4571186  0.2154260 -11.4059 < 2.2e-16 ***
## (Intercept):7   0.4968691  0.1424824   3.4872  0.000488 ***
## (Intercept):8   0.8030599  0.1709199   4.6985 2.621e-06 ***
## (Intercept):9   1.8641253  0.1799469  10.3593 < 2.2e-16 ***
## (Intercept):10 -4.1423855  0.3506563 -11.8132 < 2.2e-16 ***
## price          -6.6596694  0.1747698 -38.1054 < 2.2e-16 ***
## Income:2       -0.0042599  0.0034392  -1.2386  0.215480
## Income:3        0.0143440  0.0039221   3.6572  0.000255 ***
## Income:4        0.0040998  0.0032042   1.2795  0.200715
## Income:5       -0.0011829  0.0042971  -0.2753  0.783108
## Income:6        0.0298090  0.0047267   6.3065 2.855e-10 ***
## Income:7       -0.0092456  0.0045935  -2.0128  0.044140 *
## Income:8        0.0219965  0.0038203   5.7578 8.522e-09 ***
## Income:9        0.0169911  0.0039155   4.3394 1.428e-05 ***
## Income:10       0.0087596  0.0103007   0.8504  0.395112
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -7417.9
## McFadden R^2:  0.10475
## Likelihood ratio test : chisq = 1735.8 (p.value = < 2.22e-16)
```

```r
#removing one choice, removing choice10 data
pricer<-price %>%
  filter(price$choice!=1)
pricer<-pricer[-c(2)]
```

```
all1<-all %>%
  filter(all$choice!=1)
ni=nrow(pricer)
nj=ncol(pricer[,2:10])
Y=class.ind(all1$choice)
```

```
#computing ll
mixed_ll2<-function(para){
  beta1=matrix(0,nrow=1,ncol=10)
gamma1=matrix(0,nrow=1,ncol=18)
beta1[1,1]=0
beta1[1,2:9]=para[1:8]
beta1[1,10]=para[9]
gamma1[1,1]=0
gamma1[1,2:9]=para[1:8]
gamma1[1,10]=0
gamma1[1,11:18]=para[10:17]
intercept=cbind(0,matrix(rep(beta1[2:9],each=ni),ni,nj-1))/2
intercept1=cbind(0,matrix(rep(gamma1[2:9],each=ni),ni,nj-1))/2
intercept2=cbind(0,matrix(rep(gamma1[11:18],each=ni),ni,nj-1))
XB=intercept1+intercept2*cbind(all1[,13],all1[,13],all1[,13],all1[,13],all1[,13],all1[,13],all1[,13],al
WG=all1[,4:12]*para[9]+intercept
e=exp(XB+WG)
se=rowSums(e)
prob=e/se
ll=sum(Y*log(prob))
return(-ll)
}
```

```
set.seed(0)
model4 <-optim(runif(17,-0.1,0.1),mixed_ll2,method="BFGS", hessian = FALSE)
```

```
model4$par
```

```
##  [1]  1.640124061 -0.939582637 -1.965470973 -1.643520435  1.226333094
##  [6]  1.558823515  2.579653491 -3.240009390 -6.421848098  0.018279502
## [11]  0.007400108  0.002966837  0.033461182 -0.004447402  0.025834253
## [16]  0.020906619  0.012525749
```

```
logit01=mlogit.data(pricer,varying=2:10,shape="wide",sep="",choice="choice")
logit11=mlogit(choice~price|Income,data=logit01)
summary(logit11)
```

```
##
## Call:
## mlogit(formula = choice ~ price | Income, data = logit01, method = "nr")
##
## Frequencies of alternatives:choice
##        2        3        4        5        6        7        8        9
## 0.258506 0.089867 0.219305 0.116494 0.027367 0.117973 0.075074 0.083210
##       10
```

```
## 0.012204
##
## nr method
## 6 iterations, 0h:0m:1s
## g'(-H)^-1g = 0.000906
## successive function values within tolerance limits
##
## Coefficients :
##                   Estimate Std. Error  z-value  Pr(>|z|)
## (Intercept):3    1.6364939  0.1943569   8.4200 < 2.2e-16 ***
## (Intercept):4   -0.9428325  0.1239508  -7.6065 2.820e-14 ***
## (Intercept):5   -1.9689328  0.1514053 -13.0044 < 2.2e-16 ***
## (Intercept):6   -1.6476817  0.2327193  -7.0801 1.440e-12 ***
## (Intercept):7    1.2230785  0.1651403   7.4063 1.299e-13 ***
## (Intercept):8    1.5551291  0.2084066   7.4620 8.527e-14 ***
## (Intercept):9    2.5760219  0.2254343  11.4269 < 2.2e-16 ***
## (Intercept):10  -3.2438745  0.3515806  -9.2265 < 2.2e-16 ***
## price           -6.4220997  0.2446946 -26.2454 < 2.2e-16 ***
## Income:3         0.0184166  0.0045476   4.0497 5.127e-05 ***
## Income:4         0.0075214  0.0039386   1.9097   0.05617 .
## Income:5         0.0030968  0.0048094   0.6439   0.51964
## Income:6         0.0336065  0.0053594   6.2705 3.598e-10 ***
## Income:7        -0.0043201  0.0050706  -0.8520   0.39422
## Income:8         0.0259724  0.0044753   5.8035 6.494e-09 ***
## Income:9         0.0210461  0.0045316   4.6443 3.412e-06 ***
## Income:10        0.0126704  0.0103331   1.2262   0.22012
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -4884.2
## McFadden R^2:  0.076129
## Likelihood ratio test : chisq = 804.93 (p.value = < 2.22e-16)
```

```r
ni=nrow(price)
nj=ncol(price[,2:11])
Y=matrix(0,ni,nj)
for (i in 1:nj){
  for (j in 2:ni){
    if (price$choice[j]==i){
      Y[j,i]=1
    }
  }
}
mixed_ll(model3$par)
```

```
## [1] 7416.69
```

We can see that the diff between log likelihood reaches 3400, which shows that IIA has been violated.