Mengyang Zheng HW 2 3/17/2021

Exercise 1: Data Description

```
rm(list=ls())
#install.packages("bayesm")
#install.packages("qwraps2")
library(bayesm)
marg=data(margarine)
price=margarine$choicePrice
demo=margarine$demos
all=merge(price,demo,by="hhid")
```

```
#Average and dispersion in product characteristics
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#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))
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 by product characteristics
#Find price for each choice
all$choiceprice=0
for (i in 1:nrow(all)){
  if (all$choice[i]==1){
  all$choiceprice[i]=all[i,3]
  }
}
for (i in 1:nrow(all)){
  if (all$choice[i]==2){
  all$choiceprice[i]=all[i,4]
  }
}
for (i in 1:nrow(all)){
  if (all$choice[i]==3){
  all$choiceprice[i]=all[i,5]
  }
}
for (i in 1:nrow(all)){
  if (all$choice[i]==4){
  all$choiceprice[i]=all[i,6]
  }
}
for (i in 1:nrow(all)){
  if (all$choice[i]==5){
  all$choiceprice[i]=all[i,7]
  }
}
for (i in 1:nrow(all)){
  if (all$choice[i]==6){
  all$choiceprice[i]=all[i,8]
  }
}
for (i in 1:nrow(all)){
  if (all$choice[i]==7){
  all$choiceprice[i]=all[i,9]
  }
}
for (i in 1:nrow(all)){
  if (all$choice[i]==8){
  all$choiceprice[i]=all[i,10]
  }
}
for (i in 1:nrow(all)){
  if (all$choice[i]==9){
  all$choiceprice[i]=all[i,11]
  }
}
for (i in 1:nrow(all)){
  if (all$choice[i]==10){
  all$choiceprice[i]=all[i,12]
  }
}
```

```
#compute market share by unit and market share below/above average
des2= all %>%
  group_by(choice) %>%
  summarize(
    totalunit=n()
  ) %>%
  mutate(
    unitshare=totalunit/sum(totalunit)
  )

desx= all %>%
  group_by(choice) %>%
  summarize(
    unitbelow=sum(choiceprice<=mean(choiceprice)),
    unitabove=sum(choiceprice>mean(choiceprice))
  ) %>%
  mutate(
    belowshare=unitbelow/sum(unitbelow),
    aboveshare=unitabove/sum(unitabove)
  )
desx
```

```
## # A tibble: 10 x 5
##    choice unitbelow unitabove belowshare aboveshare
##  *  <dbl>     <int>     <int>      <dbl>      <dbl>
##  1      1       779       987    0.389      0.400
##  2      2       304       395    0.152      0.160
##  3      3       190        53    0.0948     0.0215
##  4      4       297       296    0.148      0.120
##  5      5       175       140    0.0873     0.0568
##  6      6        56        18    0.0279     0.00730
##  7      7        78       241    0.0389     0.0978
##  8      8        87       116    0.0434     0.0471
##  9      9        25       200    0.0125     0.0811
## 10     10        14        19    0.00698    0.00771
```

```
label2=colnames(all[3:12])
des2=round(des2,digits=3)
des2$choice[1:10]=label2[1:10]
desx=round(desx,digits=3)

des2=cbind(des2,desx[2:5])
des2
```

```
##      choice totalunit unitshare unitbelow unitabove belowshare aboveshare
## 1   PPk_Stk      1766     0.395       779       987      0.389      0.400
## 2   PBB_Stk       699     0.156       304       395      0.152      0.160
## 3   PFl_Stk       243     0.054       190        53      0.095      0.022
## 4  PHse_Stk       593     0.133       297       296      0.148      0.120
## 5  PGen_Stk       315     0.070       175       140      0.087      0.057
## 6  PImp_Stk        74     0.017        56        18      0.028      0.007
## 7   PSS_Tub       319     0.071        78       241      0.039      0.098
## 8   PPk_Tub       203     0.045        87       116      0.043      0.047
## 9   PFl_Tub       225     0.050        25       200      0.012      0.081
## 10 PHse_Tub        33     0.007        14        19      0.007      0.008
```

```
#Illustrate mapping between observed attributes and choices
#Ultimately we try to find most preferred choice for all dummy attributes from demo
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)
  )
notdes3= all %>%
  group_by(choice) %>%
  summarize(
    notcollege=sum(college==0),
    notwhtcollar=sum(whtcollar==0),
    notretired=sum(retired==0)
  )
des3=merge(des3,notdes3)
des3$choice[1:10]=label2[1:10]
des3
```

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

```
#As we can see from table below, all attributes tend to choose Pk_Stk as first choice and then c
hoose BB_Stk as 2nd choice except for those family size over 5 who tend to choose Hse_Stk as 2nd
choice.

#This is pretty much meaningless so we can compute the market share for each attribute and see t
he comparison difference.
des3share= des3 %>%
  summarize(
    famsize1_2=des3[1:10,2]/sum(famsize1_2),
    famsize3_4=des3[1:10,3]/sum(famsize3_4),
    famsize5.=des3[1:10,4]/sum(famsize5.),
    college=des3[1:10,5]/sum(college),
    whtcollar=des3[1:10,6]/sum(whtcollar),
    retired=des3[1:10,7]/sum(retired),
    notcollege=des3[1:10,8]/sum(notcollege),
    notwhtcollar=des3[1:10,9]/sum(notwhtcollar),
    notretired=des3[1:10,10]/sum(notretired)
  )
des3share=round(des3share,digits=3)
choice=label2[1:10]
des3share=cbind(choice,des3share)
des3share
```

```
##      choice famsize1_2 famsize3_4 famsize5. college whtcollar retired
## 1   PPk_Stk      0.370      0.413     0.401   0.397     0.388   0.364
## 2   PBB_Stk      0.155      0.165     0.129   0.155     0.146   0.174
## 3   PFl_Stk      0.096      0.028     0.033   0.078     0.051   0.134
## 4  PHse_Stk      0.105      0.136     0.195   0.123     0.135   0.094
## 5  PGen_Stk      0.039      0.086     0.104   0.061     0.087   0.048
## 6  PImp_Stk      0.020      0.008     0.038   0.023     0.016   0.029
## 7   PSS_Tub      0.085      0.072     0.033   0.073     0.071   0.049
## 8   PPk_Tub      0.042      0.056     0.018   0.037     0.045   0.021
## 9   PFl_Tub      0.087      0.031     0.018   0.044     0.050   0.084
## 10 PHse_Tub      0.002      0.005     0.030   0.011     0.012   0.004
##    notcollege notwhtcollar notretired
## 1       0.394        0.405      0.404
## 2       0.157        0.170      0.152
## 3       0.044        0.059      0.033
## 4       0.137        0.129      0.143
## 5       0.075        0.048      0.077
## 6       0.014        0.017      0.013
## 7       0.071        0.072      0.078
## 8       0.049        0.046      0.052
## 9       0.053        0.051      0.041
## 10      0.006        0.001      0.008
```

```
#As we can see from the share table, we can see the share change across different attributes. Bi
gger family size tend to have bigger share in Hse_Stk and Gen_Stk. No big difference between sha
res of college or not college and whitecollar or not whitecollar. However, for retired and non-r
etired group, nonretired group tend to buy more of Pk_Stk and Hse_Stk.
```

Exercise 2: First Model

```
#The first model we would use conditional logit since price does not vary across different house
holds.
library(mlogit)
```

```
## Loading required package: dfidx
```

```
##
## Attaching package: 'dfidx'
```

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

```
library(stargazer)
```

```
##
## 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
```

```
library(texreg)
```

```
## 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
```

```
library(survival)
library(nnet)
library(stringr)

#Use mlogit package to compute the coefficients
price=margarine$choicePrice
colnames(price)[3:12]=str_c("price",1:10)
clogit0=mlogit.data(price,varying=3:12,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:1s
## 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)
```

```
#Write the likelihood for conditional logit and optimize the model

#reference: https://stats.stackexchange.com/questions/389758/conditional-logistic-regression-in-
r

ni=nrow(price)
nj=ncol(price[,3:12])

#First we introduce a choice matrix where for individual i, if product j is chosen then cm[i,j]=
1
cm=matrix(0,ni,nj)

for (i in 1:ni){
  if (price[i,2]==1){
    cm[i,1]=1
  } else if (price[i,2]==2){
    cm[i,2]=1
  } else if (price[i,2]==3){
    cm[i,3]=1
  } else if (price[i,2]==4){
    cm[i,4]=1
  } else if (price[i,2]==5){
    cm[i,5]=1
  } else if (price[i,2]==6){
    cm[i,6]=1
  } else if (price[i,2]==7){
    cm[i,7]=1
  } else if (price[i,2]==8){
    cm[i,8]=1
  } else if (price[i,2]==9){
    cm[i,9]=1
  } else if (price[i,2]==10){
    cm[i,10]=1
  }
}


clogit_ll<-function(beta){

  #Adding 0 as the base coefficient(whenever cbind with 0 it means adding the base coefficient i
n it)
  b1=cbind(0,matrix(rep(beta[1:(nj-1)],each=ni),ni,nj-1))

  #Use the lecture definition of conditional logit to compute the likelihood
  XB=price[,3:12]*beta[nj]
  XB=b1+XB
  eXB=exp(XB)
  teXB=rowSums(eXB)
  prob=eXB/teXB

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

```
}

set.seed(0)
model1 <- optim(runif(10,-0.1,0.1),clogit_ll,method="BFGS")
model1$par
```

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

```
#Same result as the package result so it should be correct. The interpretation of the coefficien
t (-6.656) is that price is negatively related to demand. If price is high, then the product wil
l be less likely to get purchased.
```

Exercise 3: Second Model

```
#Now we can use our all data, which is the merge of choicePrice and demos.

#The second model should be the multinomial logit model since income varies across different hou
sehold id.

#Use the mlogit package to test again.
colnames(all)[3:12]=str_c("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:1s
## 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)
```

```
#Write the likelihood and optimize the model
mlogit_ll <- function(beta){

  #We now have 2 sets of coefficients, same adding 0 as the base coefficient
  b1=cbind(0,matrix(rep(beta[1:(nj-1)],each=ni),ni,nj-1))
  b2=cbind(0,matrix(rep(beta[nj:(2*(nj-1))],each=ni),ni,nj-1))

  #Since income is for each household, so same for all choices on household level
  XB=b1+b2*cbind(all[,13],all[,13],all[,13],all[,13],all[,13],all[,13],all[,13],all[,13],all[,13
],all[,13])

  #Same calculation process as model1
  eXB=exp(XB)
  teXB=rowSums(eXB)
  prob=eXB/teXB
  llik=sum(cm*log(prob))
  return(-llik)
}

set.seed(0)
model2 <- optim(runif(18,-0.1,0.1),mlogit_ll,method="BFGS")
model2$par[1:18]
```

```
##  [1] -0.843495887 -2.397478683 -1.199419682 -1.688562761 -4.137028309
##  [6] -1.529106395 -2.845978521 -2.573343270 -4.280080692 -0.003156534
## [11]  0.014502403  0.003978203 -0.001326591  0.030522991 -0.007003193
## [16]  0.022804588  0.017661189  0.010711328
```

```
#The result from second model is very close to the result from the package one. A higher income
 will have a better likelihood to purchase product 3,4,6,8,9,10 and less likely to purchase prod
uct 2,5,7 in comparison to the likelihood of product 1 (the product number is in the order of th
e column names).
```

Exercise 4: Marginal Effects

```
#Marginal effect of the first model by package
effects(clogit1,covariate="price")
```

```
##              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
```

```
#The result of marginal effect of the first model is a 10 by 10 matrix

b=as.matrix(c(0,model1$par[1:9]))
me1=matrix(0,10,10)
for (i in 1:ni){
  eXB1=as.matrix(exp(price[i,3:12]*model1$par[10]+b))
  teXB1=rowSums(eXB1)
  prob1=eXB1/teXB1
  df=matrix(0,10,10)
  indicator=diag(1,10,10)
  for (m in 1:10){
    indicator[,m]=indicator[,m]-prob1
  }
  for (n in 1:10){
    df[n,]=indicator[n,]*model1$par[10]*prob1
  }

  me1=me1+df
}

me1/ni
```

```
##              [,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
```

```
#Close to our package marginal effect result

#So from marginal effect we can tell that in comparison, purchasing more of any product will inc
rease the probability of purchasing all other products and decrease the probability of buying it
self again (Marginal diminishing law?).
```

```
#Marginal effect of the second model by package
effects(mlogit1,covariate="Income")
```

```
##             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
```

```
#Now we want to compute the marginal effect for the second model without package use
b1=c(0,model2$par[1:9])
b2=c(0,model2$par[10:18])
me2=matrix(0,1,nj)
for (i in 1:ni){
  eXB2=exp(b1+b2*cbind(all[i,13],all[i,13],all[i,13],all[i,13],all[i,13],all[i,13],all[i,13],all
[i,13],all[i,13],all[i,13]))
  teXB2=rowSums(eXB2)
  prob2=eXB2/teXB2
  me2=me2+prob2*(b2-rowSums(b2*prob2))
}
me2/ni
```

```
##               [,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
```

```
#Confirmed they have very close estimation.
#I believe increasing in income would not influence the chances of buying different choices by m
uch since all the coefficient are pretty small and income does not have a large unit (ranged fro
m 7.5 to 130).
```

Exercise 5: IIA

```
#Use package to check the coefficients before we remove one choice
mxlogit1=mlogit(choice~price|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:1s
## 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)
```

```
#Now we remove one choice from our data and rerun everything

all2=all
all2=all2[all2$choice !=10,]
mxlogit0=mlogit.data(all2,varying=3:11,shape="wide",sep="",choice="choice")
mxlogit2=mlogit(choice~price|Income,data=mxlogit0)
summary(mxlogit2)
```

```
##
## Call:
## mlogit(formula = choice ~ price | Income, data = mxlogit0, method = "nr")
##
## Frequencies of alternatives:choice
##        1        2        3        4        5        6        7        8
## 0.398017 0.157539 0.054767 0.133649 0.070994 0.016678 0.071895 0.045752
##        9
## 0.050710
##
## nr method
## 6 iterations, 0h:0m:1s
## g'(-H)^-1g = 0.000902
## successive function values within tolerance limits
##
## Coefficients :
##                 Estimate Std. Error  z-value  Pr(>|z|)
## (Intercept):2 -0.8456781  0.1037222  -8.1533 4.441e-16 ***
## (Intercept):3  0.8902279  0.1598555   5.5690 2.563e-08 ***
## (Intercept):4 -1.8272454  0.1030859 -17.7255 < 2.2e-16 ***
## (Intercept):5 -2.8738971  0.1346734 -21.3397 < 2.2e-16 ***
## (Intercept):6 -2.4555866  0.2153077 -11.4050 < 2.2e-16 ***
## (Intercept):7  0.4953162  0.1424907   3.4761 0.0005087 ***
## (Intercept):8  0.8065778  0.1713693   4.7067 2.518e-06 ***
## (Intercept):9  1.8684671  0.1807678  10.3363 < 2.2e-16 ***
## price         -6.6622619  0.1773451 -37.5667 < 2.2e-16 ***
## Income:2      -0.0041388  0.0034332  -1.2055 0.2279976
## Income:3       0.0142684  0.0039131   3.6463 0.0002660 ***
## Income:4       0.0040389  0.0031968   1.2634 0.2064330
## Income:5      -0.0012321  0.0042866  -0.2874 0.7737894
## Income:6       0.0297055  0.0047194   6.2944 3.086e-10 ***
## Income:7      -0.0092135  0.0045835  -2.0101 0.0444162 *
## Income:8       0.0218706  0.0038120   5.7373 9.618e-09 ***
## Income:9       0.0168581  0.0039069   4.3150 1.596e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -7240.2
## McFadden R^2:  0.10515
## Likelihood ratio test : chisq = 1701.6 (p.value = < 2.22e-16)
```

```
#According to the package IIA property has been rejected if we remove choice 10
hmftest(mxlogit1,mxlogit2)
```

```
##
##  Hausman-McFadden test
##
## data:  mlogit0
## chisq = -8.5483, df = 17, p-value = 1
## alternative hypothesis: IIA is rejected
```

```
mxlogit_ll <- function(beta){

  b1=cbind(0,matrix(rep(beta[1:(nj-1)],each=ni),ni,nj-1))
  b2=cbind(0,matrix(rep(beta[(nj+1):(2*nj-1)],each=ni),ni,nj-1))

  #Term of XB+WGamma on the slide
  XB=price[,3:12]*beta[nj]+b1+b2*cbind(all[,13],all[,13],all[,13],all[,13],all[,13],all[,13],all
[,13],all[,13],all[,13],all[,13])+b1
  eXB=exp(XB)
  teXB=rowSums(eXB)
  prob=eXB/teXB
  llik=sum(cm*log(prob))
  return(-llik)
}

set.seed(0)
model3=optim(runif(19,-0.1,0.1),mxlogit_ll)
bf=model3$par
bf
```

```
##  [1]  0.0986311520  0.0241609628  0.0908242807 -0.0590872300  0.0447843213
##  [6] -0.0069180636  0.0326055043  0.1025920391  0.1296103993  0.0973328100
## [11]  0.0001690686 -0.0200299436 -0.0044524869 -0.0305314506 -0.0478688011
## [16] -0.0178568978 -0.0338658613 -0.0155028148 -0.1528926833
```

```
#Since we drop choice 10 we need to re-construct the choice matrix without tenth column
cm2=matrix(0,ni,nj-1)

for (i in 1:ni){
  if (price[i,2]==1){
    cm2[i,1]=1
  } else if (price[i,2]==2){
    cm2[i,2]=1
  } else if (price[i,2]==3){
    cm2[i,3]=1
  } else if (price[i,2]==4){
    cm2[i,4]=1
  } else if (price[i,2]==5){
    cm2[i,5]=1
  } else if (price[i,2]==6){
    cm2[i,6]=1
  } else if (price[i,2]==7){
    cm2[i,7]=1
  } else if (price[i,2]==8){
    cm2[i,8]=1
  } else if (price[i,2]==9){
    cm2[i,9]=1
  }
}

cm2=cm2[!(cm2[,1]==0 & cm2[,2]==0 & cm2[,3]==0 & cm2[,4]==0 & cm2[,5]==0 & cm2[,6]==0 & cm2[,7]=
=0 & cm2[,8]==0 & cm2[,9]==0) ,]

#Then basically repeat what we did before
mxlogit_ll2 <- function(beta){
  ni=nrow(all2)
  b1=cbind(0,matrix(rep(beta[1:(nj-2)],each=ni),ni,nj-1))
  b2=cbind(0,matrix(rep(beta[nj:(nj+7)],each=ni),ni,nj-1))

  #Term of XB+WGamma on the slide
  XB=price[,3:11]*beta[nj-1]+b1+b2*cbind(all2[,13],all2[,13],all2[,13],all2[,13],all2[,13],all2
[,13],all2[,13],all2[,13],all2[,13],all2[,13])+b1
  eXB=exp(XB)
  teXB=rowSums(eXB)
  prob=eXB/teXB
  llik=sum(cm2*log(prob))
  return(-llik)
}

set.seed(0)
model4=optim(runif(17,-0.1,0.1),mxlogit_ll2)
br=model4$par
br
```

```
##  [1]  0.088197798  0.011632627 -0.025115093  0.016424721  0.085487707
##  [6] -0.060951070  0.097929447  0.085651995  0.096344625 -0.008792491
## [11] -0.036386434 -0.006130109 -0.030579630 -0.078575408 -0.025806588
## [16] -0.047886447 -0.045930804
```

```
#Reshape the full model coefficients so bf and br have the same length
bf=bf[c(1:8,10:18)]

#Compute the test statistics
MTT=c(0,-2*(bf-br))
chisq.test(abs(MTT))
```

```
## Warning in chisq.test(abs(MTT)): Chi-squared approximation may be incorrect
```

```
##
##  Chi-squared test for given probabilities
##
## data:  abs(MTT)
## X-squared = 1.217, df = 17, p-value = 1
```

```
#So we reject the IIA property as p-value is equal to 1. (I understand my results are different
 from the package results. I think this happens mostly because I somehow did not write correct m
ixed logit formula. I do not know how to correct my answer. But for the rest of the parts should
be correct including building the specification model with 1 less choice.)
```