HW03

the runtime is 41s

Jie Ren

March 5, 2019

Exercise 1: Data Description

```
rm(list=ls())
ptm <- proc.time()
# install.packages("bayesm")
# install.packages("data.table")
# install.packages("mlogit")
library("mlogit")
library("bayesm")
library("data.table")

data("margarine")
choicePrice <- margarine$choicePrice
demos       <- margarine$demos

# Mark the chosen one
choicePrice$chosen      <- colnames(choicePrice[,-(1:2)])[choicePrice$choice]
choicePrice$chosenChar  <- sapply(strsplit(choicePrice$chosen, "_"), "[[", 2)
choicePrice$chosenBrand <- sapply(strsplit(choicePrice$chosen, "_"), "[[", 1)
```

Avg and Sd of Price by characteristic

By Type

```
# Extract price data by Char
Stk <- as.matrix(choicePrice[,grepl("Stk",colnames(choicePrice))])
Tub <- as.matrix(choicePrice[,grepl("Tub",colnames(choicePrice))])

# Calculate the Avg and Sd
byType           <- data.frame(average = c(mean(Stk),mean(Tub)), sd = c(sd(Stk),sd(Tu
b)))
rownames(byType) <- c("Stk","Tub")
byType
```

```
##       average        sd
## Stk 0.6066458 0.2494704
## Tub 0.9151370 0.2448335
```

By Brand

```
# Extract price data by Brand
# Getting a list of brand
brandlist <- unique(sapply(strsplit(colnames(choicePrice)[3:12],"_"),"[[", 1))
byBrand <- data.frame(avg = numeric(7),Sd = numeric(7) ,row.names = brandlist)
for (i in 1:length(brandlist)){
  price <- as.matrix(choicePrice[,grepl(brandlist[i],colnames(choicePrice))])
  byBrand[i,]<-c(mean(price),sd(price))
}
byBrand
```

```
##            avg         Sd
## PPk  0.7979228 0.29981617
## PBB  0.5432103 0.12033186
## PFl  1.1021980 0.09284114
## PHse 0.5029105 0.11836152
## PGen 0.3452819 0.03516605
## PImp 0.7807785 0.11464607
## PSS  0.8250895 0.06121159
```

By columns

```
byCol <- data.frame(avg = apply(choicePrice[,3:12],2,mean)
,Sd = apply(choicePrice[,3:12],2,sd)
 ,row.names = colnames(choicePrice)[3:12])
byCol
```

```
##                avg         Sd
## PPk_Stk  0.5184362 0.15051740
## PBB_Stk  0.5432103 0.12033186
## PFl_Stk  1.0150201 0.04289519
## PHse_Stk 0.4371477 0.11883123
## PGen_Stk 0.3452819 0.03516605
## PImp_Stk 0.7807785 0.11464607
## PSS_Tub  0.8250895 0.06121159
## PPk_Tub  1.0774094 0.02972613
## PFl_Tub  1.1893758 0.01405451
## PHse_Tub 0.5686734 0.07245500
```

Market Share

Market Share by Brand

```
table(choicePrice$chosenBrand)/nrow(choicePrice)
```

```
## 
##        PBB        PFl       PGen       PHse       PImp        PPk 
## 0.15637584 0.10469799 0.07046980 0.14004474 0.01655481 0.44049217 
##        PSS 
## 0.07136465
```

Market Share by Char

```
table(choicePrice$chosenChar)/nrow(choicePrice)
```

```
## 
##       Stk       Tub 
## 0.8255034 0.1744966
```

Market Share by Both

```
table(choicePrice$chosen)/nrow(choicePrice)
```

```
## 
##    PBB_Stk    PFl_Stk    PFl_Tub   PGen_Stk   PHse_Stk   PHse_Tub 
## 0.15637584 0.05436242 0.05033557 0.07046980 0.13266219 0.00738255 
##   PImp_Stk    PPk_Stk    PPk_Tub    PSS_Tub 
## 0.01655481 0.39507830 0.04541387 0.07136465
```

Mapping between observed attributes and choices

```
choicePrice <- merge(choicePrice, demos, by = "hhid", all.x = TRUE) # merge choicePri
ce and demo

map <- lapply(choicePrice[,c("Income", "Fs3_4","Fs5.","Fam_Size", "college", "whtcoll
ar", "retired")],
                         function(x) xtabs(~x + choicePrice$chosen)) # Mapping using
xtab
mapShare <- sapply(map, function(x) x/rowSums(x)) # Get the market share
mapShare
```

```
## $Income
##       choicePrice$chosen
```

```
## x          PBB_Stk     PFl_Stk     PFl_Tub    PGen_Stk    PHse_Stk
##   2.5  0.080000000 0.000000000 0.040000000 0.120000000 0.040000000
##   7.5  0.183050847 0.044067797 0.074576271 0.064406780 0.115254237
##   12.5 0.214141414 0.082828283 0.050505051 0.046464646 0.088888889
##   17.5 0.147710487 0.039881832 0.029542097 0.031019202 0.163958641
##   22.5 0.145907473 0.040332147 0.035587189 0.145907473 0.182680902
##   27.5 0.197478992 0.018907563 0.071428571 0.037815126 0.140756303
##   32.5 0.153005464 0.051001821 0.060109290 0.098360656 0.116575592
##   37.5 0.121863799 0.060931900 0.032258065 0.082437276 0.103942652
##   42.5 0.108910891 0.108910891 0.046204620 0.019801980 0.075907591
##   47.5 0.117021277 0.122340426 0.010638298 0.037234043 0.085106383
##   55   0.149253731 0.054726368 0.084577114 0.034825871 0.159203980
##   67.5 0.078431373 0.019607843 0.000000000 0.117647059 0.156862745
##   87.5 0.270270270 0.081081081 0.324324324 0.000000000 0.027027027
##   130  0.038461538 0.115384615 0.192307692 0.076923077 0.307692308
##       choicePrice$chosen
## x         PHse_Tub    PImp_Stk     PPk_Stk     PPk_Tub     PSS_Tub
##   2.5  0.000000000 0.000000000 0.380000000 0.020000000 0.320000000
##   7.5  0.003389831 0.006779661 0.396610169 0.020338983 0.091525424
##   12.5 0.006060606 0.018181818 0.395959596 0.016161616 0.080808081
##   17.5 0.002954210 0.007385524 0.469719350 0.028064993 0.079763663
##   22.5 0.009489917 0.002372479 0.346381969 0.042704626 0.048635824
##   27.5 0.008403361 0.012605042 0.409663866 0.052521008 0.050420168
##   32.5 0.009107468 0.007285974 0.380692168 0.034608379 0.089253188
##   37.5 0.017921147 0.003584229 0.473118280 0.050179211 0.053763441
##   42.5 0.003300330 0.066006601 0.412541254 0.069306931 0.089108911
##   47.5 0.015957447 0.090425532 0.441489362 0.047872340 0.031914894
##   55   0.000000000 0.014925373 0.233830846 0.208955224 0.059701493
##   67.5 0.019607843 0.039215686 0.372549020 0.058823529 0.137254902
##   87.5 0.000000000 0.027027027 0.243243243 0.000000000 0.027027027
##   130  0.000000000 0.076923077 0.192307692 0.000000000 0.000000000
## 
## $Fs3_4
##    choicePrice$chosen
## x       PBB_Stk     PFl_Stk     PFl_Tub    PGen_Stk    PHse_Stk
##   0 0.148423818 0.079246935 0.068739054 0.056042032 0.129159370
##   1 0.164684355 0.028362306 0.031107045 0.085544373 0.136322049
##    choicePrice$chosen
## x      PHse_Tub    PImp_Stk     PPk_Stk     PPk_Tub     PSS_Tub
##   0 0.009194396 0.024518389 0.378283713 0.035464098 0.070928196
##   1 0.005489478 0.008234218 0.412625801 0.055809698 0.071820677
## 
## $Fs5.
##    choicePrice$chosen
## x       PBB_Stk     PFl_Stk     PFl_Tub    PGen_Stk    PHse_Stk
##   0 0.160631143 0.057682359 0.055354371 0.065183652 0.122866011
##   1 0.129139073 0.033112583 0.018211921 0.104304636 0.195364238
##    choicePrice$chosen
## x      PHse_Tub    PImp_Stk     PPk_Stk     PPk_Tub     PSS_Tub
```

```
##    0 0.003879979 0.013191930 0.394205898 0.049663735 0.077340921
##    1 0.029801325 0.038079470 0.400662252 0.018211921 0.033112583
##
## $Fam_Size
##    choicePrice$chosen
## x       PBB_Stk     PFl_Stk     PFl_Tub    PGen_Stk    PHse_Stk
##   1 0.139204545 0.107954545 0.096590909 0.028409091 0.065340909
##   2 0.159638554 0.092620482 0.084337349 0.041415663 0.115963855
##   3 0.172233820 0.030271399 0.050104384 0.062630480 0.124217119
##   4 0.158794788 0.026872964 0.016286645 0.103420195 0.145765472
##   5 0.134177215 0.050632911 0.027848101 0.083544304 0.182278481
##   6 0.121546961 0.000000000 0.000000000 0.132596685 0.182320442
##   7 0.083333333 0.000000000 0.000000000 0.166666667 0.666666667
##   8 0.125000000 0.000000000 0.000000000 0.250000000 0.312500000
##    choicePrice$chosen
## x      PHse_Tub    PImp_Stk      PPk_Stk     PPk_Tub     PSS_Tub
##   1 0.000000000 0.019886364 0.420454545 0.051136364 0.071022727
##   2 0.002259036 0.019578313 0.356927711 0.039156627 0.088102410
##   3 0.003131524 0.011482255 0.417536534 0.048016701 0.080375783
##   4 0.007328990 0.005700326 0.408794788 0.061889251 0.065146580
##   5 0.032911392 0.058227848 0.405063291 0.005063291 0.020253165
##   6 0.027624309 0.000000000 0.419889503 0.049723757 0.066298343
##   7 0.000000000 0.000000000 0.083333333 0.000000000 0.000000000
##   8 0.000000000 0.000000000 0.312500000 0.000000000 0.000000000
##
## $college
##    choicePrice$chosen
## x       PBB_Stk     PFl_Stk     PFl_Tub    PGen_Stk    PHse_Stk
##   0 0.157068063 0.043520942 0.053337696 0.074934555 0.137107330
##   1 0.154879774 0.077793494 0.043847242 0.060820368 0.123055163
##    choicePrice$chosen
## x      PHse_Tub    PImp_Stk      PPk_Stk     PPk_Tub     PSS_Tub
##   0 0.005890052 0.013743455 0.394306283 0.049410995 0.070680628
##   1 0.010608204 0.022630835 0.396746818 0.036775106 0.072842999
##
## $whtcollar
##    choicePrice$chosen
## x       PBB_Stk     PFl_Stk     PFl_Tub    PGen_Stk    PHse_Stk
##   0 0.170405983 0.059294872 0.050747863 0.048076923 0.129273504
##   1 0.146266359 0.050808314 0.050038491 0.086605081 0.135103926
##    choicePrice$chosen
## x      PHse_Tub    PImp_Stk      PPk_Stk     PPk_Tub     PSS_Tub
##   0 0.001068376 0.017094017 0.405448718 0.046474359 0.072115385
##   1 0.011932256 0.016166282 0.387605851 0.044649731 0.070823711
##
## $retired
##    choicePrice$chosen
## x       PBB_Stk     PFl_Stk     PFl_Tub    PGen_Stk    PHse_Stk
##   0 0.151541096 0.032534247 0.041095890 0.076769406 0.143264840
```

```
##    1 0.173913043 0.133540373 0.083850932 0.047619048 0.094202899
##     choicePrice$chosen
## x       PHse_Tub    PImp_Stk     PPk_Stk     PPk_Tub     PSS_Tub
##    0 0.008276256 0.013127854 0.403538813 0.052226027 0.077625571
##    1 0.004140787 0.028985507 0.364389234 0.020703934 0.048654244
```

Exercise 2: First Model

This is a condiciotnal logit model, as price is alternative specific.

Manually

```
n   <- nrow(choicePrice)
b   <- rep(-1,10)

LL.2 <- function(b,Predict = F){
  c  <- cbind(0, t(replicate(n,b[1:9]))) # Calculate the constants
  Xb <- as.matrix(choicePrice[,3:12])*b[10] # Calculate latent utility for alternativ
e specific char
  XB <- Xb + c # Calculate latent utility
  P  <- exp(XB)/rowSums(exp(XB)) # Calculate probability
  LL <- sum(-log(P[cbind(seq(n),choicePrice$choice)])) # Only use the prob for choice
that is selected
  ifelse(Predict == F, return(LL), return(P)) # To allow the output of the probabilit
y matrix when Predit = T
}

result.2 <- optim(par = b, LL.2)
result.2$par
```

```
##  [1] -0.7539690  1.5021992 -1.6159214 -2.9593816 -1.0913599  0.2050317
##  [7]  1.6467839  2.3765521 -3.8519185 -6.7023977
```

```
result.2$value
```

```
## [1] 7486.294
```

Check with mlogit

```r
choicePrice.n <- data.frame(choicePrice)
setnames(choicePrice.n, old = c("PPk_Stk", "PBB_Stk", "PFl_Stk", "PHse_Stk", "PGen_St
k", "PImp_Stk", "PSS_Tub", "PPk_Tub", "PFl_Tub", "PHse_Tub"), new = c("Price1","Price
2","Price3","Price4","Price5","Price6","Price7","Price8","Price9","Price10")) # renam
e the column names to allow reshaping

# Reshape the data for mlogit function
Ch <- mlogit.data(choicePrice.n, shape = "wide", varying = 3:12, choice = "choice", s
ep = "", alt.levels = 1:10)

# Regress using the mlogit function
result.2.m <- mlogit(choice ~ Price, data = Ch, method = "nr")
summary(result.2.m)
```

```
##
## Call:
## mlogit(formula = choice ~ Price, data = Ch, method = "nr")
##
## Frequencies of alternatives:
##         1         2         3         4         5         6         7
## 0.3950783 0.1563758 0.0543624 0.1326622 0.0704698 0.0165548 0.0713647
##         8         9        10
## 0.0454139 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|)
## 2:(intercept)  -0.954307   0.050046 -19.0685 < 2.2e-16 ***
## 3:(intercept)   1.296968   0.108651  11.9370 < 2.2e-16 ***
## 4:(intercept)  -1.717332   0.054158 -31.7096 < 2.2e-16 ***
## 5:(intercept)  -2.904005   0.071461 -40.6379 < 2.2e-16 ***
## 6:(intercept)  -1.515311   0.126230 -12.0043 < 2.2e-16 ***
## 7:(intercept)   0.251768   0.079164   3.1803  0.001471 **
## 8:(intercept)   1.464868   0.118047  12.4092 < 2.2e-16 ***
## 9:(intercept)   2.357505   0.133774  17.6230 < 2.2e-16 ***
## 10:(intercept) -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)
```

Interpretation: The signs on the intercepts indicating that, holding other variables constant, people's general preference (+:prefer; -: not prefer) over that choice compared to the reference choice, which is PPk_Stk.
The negative sign on the price coefficient indicating that as the price of one choice increases, the individual is less likly to buy that choice (Holding other variables constant).

Exercise 3: Second Model

This is a multinomial logit model, as income is individual specific.

Mannually

```
b <- c(-1,-2,-1,-2,-4,-1,-3,-2,-4,rep(0,9)) # PPK_Stk as reference

LL.3 <- function(b,Predict = F){
  c  <- cbind(0, t(replicate(n,b[1:9]))) # Calculate the constants
  Xb <- cbind(0, t(replicate(n,b[10:18])))*choicePrice$Income # Calculate latent util
ity for individual specific char
  XB <- Xb+c # Calculate latent utility
  P  <- exp(XB)/rowSums(exp(XB)) # Calculate probability
  LL <- sum(-log(P[cbind(seq(n),choicePrice$choice)])) # Only use the prob for choice
that is selected
  ifelse(Predict == F, return(LL), return(P))
}
result.3 <- optim(par = b, LL.3)
result.3$par
```

```
##  [1] -0.6869351117 -2.0701660438 -0.9987654551 -1.4928755533 -3.9028707170
##  [6] -1.1239676158 -2.8393017277 -2.4470402370 -4.2454577722 -0.0059463453
## [11]  0.0075277238 -0.0001509262 -0.0056968769  0.0263947649 -0.0176220806
## [16]  0.0223534519  0.0144906320  0.0097841836
```

```
result.3$value
```

```
## [1] 8246.721
```

check with mlogit

```
result.3.m <- mlogit(choice ~ 0 | Income, data = Ch, method = "nr")
summary(result.3.m)
```

```
##
## Call:
## mlogit(formula = choice ~ 0 | Income, data = Ch, method = "nr")
##
## Frequencies of alternatives:
##         1         2         3         4         5         6         7
## 0.3950783 0.1563758 0.0543624 0.1326622 0.0704698 0.0165548 0.0713647
##         8         9        10
## 0.0454139 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|)
## 2:(intercept)  -0.8453241  0.0931354  -9.0763 < 2.2e-16 ***
## 3:(intercept)  -2.3998575  0.1335802 -17.9657 < 2.2e-16 ***
## 4:(intercept)  -1.2013265  0.0971021 -12.3718 < 2.2e-16 ***
## 5:(intercept)  -1.6905817  0.1269952 -13.3122 < 2.2e-16 ***
## 6:(intercept)  -4.1397653  0.2109890 -19.6208 < 2.2e-16 ***
## 7:(intercept)  -1.5310415  0.1280434 -11.9572 < 2.2e-16 ***
## 8:(intercept)  -2.8483522  0.1393848 -20.4352 < 2.2e-16 ***
## 9:(intercept)  -2.5755972  0.1361400 -18.9187 < 2.2e-16 ***
## 10:(intercept) -4.2822699  0.3457920 -12.3839 < 2.2e-16 ***
## 2:Income       -0.0030887  0.0031140  -0.9919 0.3212477
## 3:Income        0.0145862  0.0038255   3.8129 0.0001373 ***
## 4:Income        0.0040504  0.0030926   1.3097 0.1902878
## 5:Income       -0.0012536  0.0042024  -0.2983 0.7654694
## 6:Income        0.0306120  0.0046740   6.5494 5.775e-11 ***
## 7:Income       -0.0069326  0.0044161  -1.5698 0.1164518
## 8:Income        0.0228862  0.0036217   6.3192 2.629e-10 ***
## 9:Income        0.0177430  0.0037623   4.7160 2.405e-06 ***
## 10:Income       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)
```

Interpretation: (Holding other variables constant) The signs on the intercepts indicating that, holding other variables constant, people's general preference (+:prefer; -: not prefer) over that choice compared to the reference choice, which is PPk_Stk.
Beta coefficient:
2:Income -0.0030887: More income, less likely to choose choice 2 over choice 1.
3:Income 0.0145862: More income, more likely to choose choice 3 over choice 1.

4:Income 0.0040504: More income, more likely to choose choice 4 over choice 1.
5:Income -0.0012536: More income, less likely to choose choice 5 over choice 1.
6:Income 0.0306120: More income, more likely to choose choice 6 over choice 1.
7:Income -0.0069326: More income, less likely to choose choice 7 over choice 1.
8:Income 0.0228862: More income, more likely to choose choice 8 over choice 1.
9:Income 0.0177430: More income, more likely to choose choice 9 over choice 1.
10:Income 0.0107909: More income, more likely to choose choice 10 over choice 1.

Exercise 4: Marginal Effects

Marginal Effect for Conditional Logit

$$\frac{\partial p_{ij}}{\partial x_{ik}} = p_{ij}(\delta_{ijk} - p_{ik})\beta$$

```
Pij <- LL.2(result.2$par, Predict = T) # output the probability matrix at optimized b
eta
# Average Marginal effect
Marginal.C <- matrix(0,10,10)
for (j in 1:10){
  for(k in 1:10){
    delta <- ifelse(j == k, 1, 0)
    Marginal.C[j,k] <- mean(Pij[,j]*(delta-Pij[,k])*result.2$par[10])
  }
}
Marginal.C
```

```
##              [,1]        [,2]        [,3]        [,4]        [,5]
##  [1,] -1.25868739  0.311476148  0.128393202  0.29117679  0.132348010
##  [2,]  0.31147615 -0.813719253  0.068689230  0.15210471  0.072236980
##  [3,]  0.12839320  0.068689230 -0.376914414  0.05866559  0.030416047
##  [4,]  0.29117679  0.152104706  0.058665591 -0.73659173  0.059129583
##  [5,]  0.13234801  0.072236980  0.030416047  0.05912958 -0.387574420
##  [6,]  0.04984952  0.026097442  0.011213914  0.02411775  0.010976139
##  [7,]  0.12923709  0.068231451  0.029224028  0.05863711  0.030173012
##  [8,]  0.10304066  0.055141719  0.024251525  0.04448637  0.024656553
##  [9,]  0.09731001  0.052239018  0.022674101  0.04229289  0.023695774
## [10,]  0.01585598  0.007502558  0.003386774  0.00598094  0.003942323
##               [,6]         [,7]         [,8]         [,9]        [,10]
##  [1,]  0.049849517  0.129237087  0.103040659  0.097310007  0.015855977
##  [2,]  0.026097442  0.068231451  0.055141719  0.052239018  0.007502558
##  [3,]  0.011213914  0.029224028  0.024251525  0.022674101  0.003386774
##  [4,]  0.024117751  0.058637112  0.044486371  0.042292887  0.005980940
##  [5,]  0.010976139  0.030173012  0.024656553  0.023695774  0.003942323
##  [6,] -0.150325018  0.010667552  0.008358536  0.007953811  0.001090356
##  [7,]  0.010667552 -0.378096704  0.025180878  0.023032642  0.003712941
##  [8,]  0.008358536  0.025180878 -0.308526030  0.020208144  0.003201645
##  [9,]  0.007953811  0.023032642  0.020208144 -0.292438132  0.003031748
## [10,]  0.001090356  0.003712941  0.003201645  0.003031748 -0.047705262
```

Each unit increase in price of an alternative decrease the probability of selecting that alternative and increases the probability of the other alternatives, by certain percent.

Marginal Effect for Multinomial Logit

$$\frac{\partial p_{ij}}{\partial x_i} = p_{ij}(\beta_j - \bar{\beta}_i)$$

$$\bar{\beta}_i = \sum_l p_{il}\beta_l$$

```
Pij <- LL.3(result.3$par, Predict = T)
# Average Marginal effect
Marginal.M <- NULL
beta.avg <- Pij %*% c(0,result.3$par[10:18])

for (j in 1:10){
  Marginal.M[j] <- mean(Pij[,j]*(c(0,result.3$par[10:18])[j]-beta.avg))
}
Marginal.M
```

```
##  [1]  4.835376e-05 -9.188830e-04  4.379739e-04 -2.605439e-06 -3.953643e-04
##  [6]  4.453585e-04 -1.337209e-03  9.474912e-04  7.061107e-04  6.877356e-05
```

Each unit increase in the income increases/decreases (as the sign) the probability of selecting alternative j by certain percent.

Exercise 5: IIA

Mixed logit on income and price (Manually)

```
bf   <- c(-1,1,-2,-3,-2,0,1,2,-4,rep(0,9),-6)

LL.5  <- function(bf){
  c   <- cbind(0, t(replicate(n,bf[1:9]))) # Calculate the constants
  Xb2 <- cbind(0, t(replicate(n,bf[10:18])))*choicePrice$Income # Calculate latent ut
ility for individual specific char
  Xb1 <- as.matrix(choicePrice[,3:12])*bf[19] # Calculate latent utility for alternat
ive specific char
  XB  <- Xb1 + Xb2 + c # Calculate latent utility
  P   <- exp(XB)/rowSums(exp(XB)) # Calculate probability
  LL  <- sum(-log(P[cbind(seq(n),choicePrice$choice)])) # Only use the prob for choic
e that is selected
  return(LL)
}

result.5 <- optim(par = bf, LL.5)
result.5$par
```

```
##  [1] -1.0875265467  1.2213871996 -1.7937029035 -2.6293289894 -2.3978529949
##  [6]  0.2448235630  1.1521560286  2.2290273368 -3.2639521180  0.0025758995
## [11] -0.0010572558  0.0015117170 -0.0094395550  0.0211795664 -0.0056762403
## [16]  0.0050433592 -0.0006691237 -0.0217515541 -6.2991741616
```

```
result.5$value
```

```
## [1] 7458.295
```

Check with mlogit package

```
result.5.m <- mlogit(choice ~ Price | Income, data = Ch, method = "nr")
summary(result.5.m)
```

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

MTT mannually

Take out alternative 10

```
choicePrice.alt <- data.frame(choicePrice)[choicePrice$choice!=10,]

bf.alt<- c(-1,1,-2,-3,-2,0,1,2,rep(0,8),-6)
n.alt <- nrow(choicePrice.alt)
LL.5.alt <- function(bf){
  c   <- cbind(0, t(replicate(n.alt,bf[1:8]))) # Calculate the constants
  Xb2 <- cbind(0, t(replicate(n.alt,bf[9:16])))*choicePrice.alt$Income # Calculate la
tent utility for individual specific
  Xb1 <- as.matrix(choicePrice.alt[,3:11])*bf[17] # Calculate latent utility for alte
rnative specific char
  XB  <- Xb1 + Xb2 + c # Calculate latent utility
  P   <- exp(XB)/rowSums(exp(XB)) # Calculate probability
  LL  <- sum(-log(P[cbind(seq(n.alt),choicePrice.alt$choice)])) # Only use the prob f
or choice that is selected
  return(LL)
}

result.5.alt <- optim(par = bf.alt, LL.5.alt)
result.5.alt$par
```

```
##  [1] -0.7076065721  1.0717473638 -1.7212322944 -2.5432964824 -2.5532061680
##  [6]  0.1879850922  0.7309075789  1.8651078105 -0.0079205744 -0.0001850738
## [11]  0.0018372490 -0.0096570938  0.0256157658 -0.0032820676  0.0147494382
## [16]  0.0056166522 -6.0763481983
```

```
result.5.alt$value
```

```
## [1] 7259.156
```

Test statistic for MTT test

```
MTT <- 2*(LL.5.alt(result.5$par[c(1:8,10:17,19)]) - LL.5.alt(result.5.alt$par))
MTT
```

```
## [1] 36.45451
```

```
pchisq(MTT,df = length(result.5.alt$par),lower.tail = F)
```

```
## [1] 0.003986694
```

From the p-value, we can’t reject the null hypothese and state that IIA is hold.

Check IIA test by hmftest

```
result.5.m.alt <- mlogit(choice ~ Price | Income, data = Ch, method = "nr", alt.subse
t = c("1","2","3","4","5","6","7","8","9"))
# summary(result.5.m.alt)
hmftest(result.5.m, result.5.m.alt)
```

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

```
# check run time
runTime = proc.time()-ptm
```