Assignment 3

Nond Prueksiri

Estimated run time: 5 minutes

Data Preparation

```
rm(list=ls())
library(foreign)
library(weights)
library(ggplot2)
library(data.table)
library(dummies)
library(bayesm)
library(dplyr)

data(margarine)
dfdemo <- as.data.frame(margarine$demos)
dfprod <- as.data.frame(margarine$choicePrice)
```

Excercise 1

Average and dispersion in product characteristics

```
# Packaging
stick <- list(dfprod$PPk_Stk, dfprod$PBB_Stk, dfprod$PFl_Stk,
              dfprod$PHse_Stk, dfprod$PGen_Stk, dfprod$PImp_Stk)
tub   <- list(dfprod$PSS_Tub, dfprod$PPk_Tub, dfprod$PFl_Tub, dfprod$PHse_Tub)

# Brands
parkay        <-list(dfprod$PPk_Stk, dfprod$PPk_Tub)
bluebonnett   <- list(dfprod$PBB_Stk)
fleischmanns  <- list(dfprod$PFl_Stk, dfprod$PFl_Tub)
house         <- list(dfprod$PHse_Stk, dfprod$PHse_Tub)
generic       <- list(dfprod$PGen_Stk)
imperial      <- list(dfprod$PImp_Stk)
shedspread    <- list(dfprod$PSS_Tub)

ans_1a <- round(colMeans(dfprod), digits = 4)
ans_1a <- cbind(ans_1a, round(sapply(dfprod, sd),digits = 4 ))
ans_1a <- ans_1a[3:12, ]
colnames(ans_1a) <-  c("average", "dispersion (SD)")
ans_1a
```

```
##          average dispersion (SD)
## PPk_Stk   0.5184          0.1505
## PBB_Stk   0.5432          0.1203
## PFl_Stk   1.0150          0.0429
## PHse_Stk  0.4371          0.1188
## PGen_Stk  0.3453          0.0352
## PImp_Stk  0.7808          0.1146
```

```
## PSS_Tub   0.8251         0.0612
## PPk_Tub   1.0774         0.0297
## PFl_Tub   1.1894         0.0141
## PHse_Tub  0.5687         0.0725
```

Market share in product and in product characteristics

```r
dfprod$sales <-  ifelse(dfprod$choice == 1, dfprod$PPk_Stk,
                 ifelse(dfprod$choice == 2, dfprod$PBB_Stk,
                 ifelse(dfprod$choice == 3, dfprod$PFl_Stk,
                 ifelse(dfprod$choice == 4, dfprod$PHse_Stk,
                 ifelse(dfprod$choice == 5, dfprod$PGen_Stk,
                 ifelse(dfprod$choice == 6, dfprod$PImp_Stk,
                 ifelse(dfprod$choice == 7, dfprod$PSS_Tub,
                 ifelse(dfprod$choice == 8, dfprod$PPk_Tub,
                 ifelse(dfprod$choice == 9, dfprod$PFl_Tub,
                        dfprod$PHse_Tub)
                 ))))))))

pk_stk <- sum(dfprod$sales[dfprod$choice == 1])*100 / sum(dfprod$sales)
share <- pk_stk
share <- cbind(share,sum(dfprod$sales[dfprod$choice == 2])*100 / sum(dfprod$sales),
               sum(dfprod$sales[dfprod$choice == 3])*100 / sum(dfprod$sales),
               sum(dfprod$sales[dfprod$choice == 4])*100 / sum(dfprod$sales),
               sum(dfprod$sales[dfprod$choice == 5])*100 / sum(dfprod$sales),
               sum(dfprod$sales[dfprod$choice == 6])*100 / sum(dfprod$sales),
               sum(dfprod$sales[dfprod$choice == 7])*100 / sum(dfprod$sales),
               sum(dfprod$sales[dfprod$choice == 8])*100 / sum(dfprod$sales),
               sum(dfprod$sales[dfprod$choice == 9])*100 / sum(dfprod$sales),
               sum(dfprod$sales[dfprod$choice == 10])*100 / sum(dfprod$sales)
                )
sum(share)
```

```
## [1] 100
```

```r
colnames(share) <- c("pk_stk", "bb_stk", "fl_stk", "hse_stk", "gen_stk",
                     "imp_stk", "ss_tub", "pk_tub", "fl_tub" ,"hse_tub")

share
```

```
##        pk_stk   bb_stk   fl_stk  hse_stk  gen_stk  imp_stk   ss_tub
## [1,] 31.64004 12.30866 9.887261 9.316123 4.474122 2.247123 9.984262
##        pk_tub   fl_tub   hse_tub
## [1,] 8.753436 10.75665 0.6323178
```

- By type of packaging

```r
share_type <- c(sum(share[,1:6]), sum(share[,7:10]))
share_type <- rbind(c("stick", "tub") , share_type)

share_type
```

```
##            [,1]               [,2]
##            "stick"            "tub"
## share_type "69.8733351848051" "30.1266648151949"
```

Mapping Data

```
dfmar <- merge(dfprod, dfdemo, by = "hhid", all.x = TRUE)
```

Exercise 2

Model specification for effect of price on demand

Since, prices of product (denoted matrix Z) vary among choices i - 1,2,...,10 therefore, I propose that conditional logit model is to be used here. Therefore, the probability function can be written as

$$p_{ij} = \frac{exp(Z_{ij}\gamma)}{\sum_{l=1}^{m} exp(Z_{il}\gamma)}$$

for j = 1, ..., m

where gamma is a coefficient corresponding to prices However, we want to include the constant term for each choice (denoted by alpha_j), this conditional logit turns to be mixed logit

$$p_{ij} = \frac{exp(Z_{ij}\gamma + d_{ij}\alpha_j)}{\sum_{l=1}^{m} exp(Z_{il}\gamma + d_{il}\alpha_l)}$$

for j = 1, ..., m

Where d_ij = 1 for family i choosing choice j and alpha_j is a constant coefficient corresponding to choice j Note that we use choice 1 for the baseline that is alpha_1 = 0

Likelihood Function for the model

- Create matrix of prices Z (N X 10) and indicator matrix D (denoted 1 if the choice is selected, otherwise 0)

```
Z <- as.matrix(dfmar[,3:12])
D <- dummy("choice", data = dfmar, sep="")
```

- The negative log-likelihood function The function is a function of coeff where the vector coeff is

$$\begin{bmatrix} \gamma \\ 0 \\ \alpha_2 \\ ... \\ \alpha_{10} \end{bmatrix}$$

The function operates as follows;

- Assign gamma and alpha into the vector of coefficients to be estimated
- Multiply each price with gamma (scalar)
- Calculate general exponents of
$$exp(Z_{ij}\gamma + d_{ij}\alpha_j)$$
- Construct a matrix of exponents over the sum of exponents by row
- Use indicator matrix D to limit the numerator for only the choice choosen
- Create the likelihood function by summing log of all rows
- Return negative likelihood function

```
llclogit <- function(coeff) {
  gamma <- coeff[1]
  alpha <- as.matrix(coeff[2:11])
  alpha[1] <- 0
  ZG <- Z*gamma
  expZBG <- exp(ZG + (matrix(1,nrow(dfmar)) %*% t(alpha)))
  plogit <- t(apply(expZBG, 1, function(x) x / sum(x)))
  plogit <- plogit*D
  ll <- sum(log(rowSums(plogit)))
  return(-ll)
}
```

- Optimization Package nlm is used here to find the vector of coefficients "coeff" with intitial value of 0 for all of the coefficients

```
gamma <- matrix(0,nrow = 11)

clogit <- nlm(llclogit, gamma)
```

Report and Interpret the coefficient corresponding prices

The following report the coefficients.

```
clogit_coeff <- as.matrix(clogit$estimate)
row.names(clogit_coeff) <- c("Price", "Choice 1" , "Choice 2",
                             "Choice 3", "Choice 4","Choice 5",
                             "Choice 6", "Choice 7", "Choice 8",
                             "Choice 9", "Choice 10")
clogit_coeff <- round(clogit_coeff, digits = 4)
clogit_coeff
```

```
##              [,1]
## Price     -6.6566
## Choice 1   0.0000
## Choice 2  -0.9543
## Choice 3   1.2970
## Choice 4  -1.7173
## Choice 5  -2.9040
## Choice 6  -1.5153
## Choice 7   0.2518
## Choice 8   1.4649
## Choice 9   2.3575
## Choice 10 -3.8966
```

The coefficient on price is -6.6566. At this stage, we can only say that the higher the price of product the less probability that such product will be choosen. However, we cannot interpret anything about the magnitude of probability decrease as we need marginal effects to do so.

Exercise 3

Model specification for effect of family income on demand

Since, family income varies among individuals but not the alternatives therefore, I propose that multinomial logit model is to be used here. Therefore, the probability function can be written as

$$p_{ij} = \frac{exp(X_i\beta_j)}{\sum_{l=1}^{m} exp(X_i\beta_l)}$$

for j = 1, ..., m

Where matrix X is a 7 X N matrix with each column contains the characteristic of each household including income and the last column is 1 which is intended for the constant term.

Likelihood Function for the model

- Create matrix of prices X (N X 7) and indicator matrix D (denoted 1 if the choice is selected, otherwise 0)

```
X <- as.matrix(cbind(matrix(1,nrow(dfmar)), dfmar$Income,
                     dfmar$Fs3_4, dfmar$Fs5.,dfmar$college,
                     dfmar$whtcollar, dfmar$retired))
```

- The negative log-likelihood function The function is a function of coeff where the vector matrix (7 X 10) is

$$\begin{bmatrix} \beta_{1,1} & \beta_{1,2} & \dots & \beta_{1,10} \\ \beta_{2,1} & \beta_{2,2} & \dots & \beta_{2,10} \\ \dots & \dots & \dots & \dots \\ \beta_{7,1} & \beta_{7,1} & \dots & \beta_{7,10} \end{bmatrix}$$

where beta_k,j corresponds k characteristic of the family for choice j and where j = 1,2, ... , m choices and k = 1,2, ... , K characteristics

The function operates as follows;

- Assign beta as a matrix of coefficients as structured above
- Set the first choice as baseline by restricting beta_1,k = 0
- Calculate general exponents of

$$exp(X_i\beta_j)$$

- Construct a matrix of exponents over the sum of exponents by row
- Use indicator matrix D to limit the numerator for only the choice choosen
- Create the likelihood function by summing log of all rows
- Return negative likelihood function

```
llmlogit <- function(coeff) {
  beta <- matrix(coeff, nrow = 7, byrow = TRUE)
  beta[,1] <- 0
  expXB <- exp(X %*% beta)
  plogit <- t(apply(expXB, 1, function(x) x/ sum(x)))
  plogit <- plogit*D
  ll <- sum(log(rowSums(plogit)))
  return(-ll)
}

beta <- matrix(0,70)
```

- Optimization Package nlm is used here to find the vector of coefficients “coeff” with intitial value of 0 for all of the coefficients

```
beta <- matrix(0, nrow = 70)
mlogit <- nlm(llmlogit,beta)
```

Report and Interpret the coefficient corresponding prices

The following report the coefficients, since choice 1 is the baseline, thus, all coefficients are set to 0 and will not be shown

```
mlogit_coeff <- matrix(mlogit$estimate, nrow = 7, byrow = TRUE)
mlogit_coeff <- round(mlogit_coeff, digits=4)
colnames(mlogit_coeff) <- c("Choice 1" , "Choice 2", "Choice 3",
                            "Choice 4" , "Choice 5", "Choice 6",
                            "Choice 7", "Choice 8", "Choice 9",
                            "Choice 10")
row.names(mlogit_coeff) <- c("Constant", "Income", "Family Size 3-4",
                             "Family Size 5", "college", "white collar",
                             "retired" )
mlogit_coeff[,2:10]
```

```
##                 Choice 2 Choice 3 Choice 4 Choice 5 Choice 6 Choice 7
## Constant         -0.8356  -3.2160  -1.0575  -2.3547  -3.5173  -0.8201
## Income           -0.0021   0.0236   0.0038  -0.0079   0.0298  -0.0079
## Family Size 3-4  -0.0122  -1.0014  -0.0798   0.6120  -1.4359  -0.6045
## Family Size 5    -0.2732  -1.2098   0.3572   0.8296   0.2763  -1.5258
## college           0.0262   0.5159  -0.2529  -0.3488   0.2505   0.0778
## white collar     -0.0502   0.5860  -0.0114   0.6761  -0.5246  -0.0905
## retired           0.1342   1.4596  -0.3572   0.1400   0.3061  -1.0818
##                 Choice 8 Choice 9 Choice 10
## Constant         -2.0928  -2.3240   -3.6687
## Income            0.0267   0.0265   -0.0062
## Family Size 3-4  -0.4080  -1.3327   -0.9183
## Family Size 5    -1.4077  -1.9471    1.0328
## college          -0.3965  -0.3344    0.1234
## white collar     -0.3099   0.3662    0.1882
## retired          -1.3435   0.4422   -1.2139
```

Interpretation: for a family is more likely to buy product of choice 3, 4, 6, 8 and 9 than to buy choice 1 given a family earn more income while less likely to buy product of choice 2, 5, 7 and 10 than to buy choice 1

Exercise 4 Marginal Effects

The first model (Conditional Logit)

- Assign the matrix of coefficients again

```
gamma <- clogit_coeff[1,]
alpha <- c(0,clogit_coeff[2:10,])
```

- Create individual likelihood function

```
ZG <- Z* gamma
expZG <- exp(ZG + (matrix(1,nrow(ZG)) %*% t(alpha)))
pclogit <- t(apply(expZG, 1, function(x) x / sum(x)))
```

- Calculate

$$sum_{ij} p_{ij} * \gamma$$

```r
gammap <- gamma * colSums(pclogit)
```

- Calculate

$$\sum_{ijk} (p_{ij} * \gamma) * \delta_{ijk} * \alpha)$$

```r
## Write the row repclication function
rowrep<-function(x,n){
  matrix(rep(x,each=n),nrow=n)
}
## Replicate the row to make matrix conform
gammap2 <- rowrep(gammap,n=10) * diag(10)
gammap2 <- alpha * (t(pclogit) %*% pclogit)
```

- Calculate average marginal effects

$$\frac{\sum_{ijk}(p_{ij} * \gamma) * \delta_{ijk} * \alpha) - (p_{ij} * \gamma)}{N}$$

```r
mfx_clogit <- round((gammap2 - gammap)/nrow(Z), digits = 4)
colnames(mfx_clogit) <- c("Choice 1" , "Choice 2", "Choice 3",
                          "Choice 4" , "Choice 5", "Choice 6",
                          "Choice 7", "Choice 8", "Choice 9",
                          "Choice 10")
row.names(mfx_clogit) <- c("p1", "p2" ,"p3" ,"p4" ,"p5", "p6"
                         , "p7", "p8", "p9" , "p10")
mfx_clogit
```

```
##     Choice 1 Choice 2 Choice 3 Choice 4 Choice 5 Choice 6 Choice 7
## p1    0.5379   0.5379   0.5379   0.5379   0.5379   0.5379   0.5379
## p2    0.4343   0.4343   0.4343   0.4343   0.4343   0.4343   0.4343
## p3    0.0055   0.0055   0.0055   0.0052   0.0055   0.0055   0.0055
## p4    2.5631   2.5562   2.5259   2.7520   2.5409   2.5258   2.5264
## p5    0.2140   0.2142   0.2180   0.1977   0.2159   0.2180   0.2179
## p6    0.0039   0.0040   0.0041   0.0034   0.0040   0.0041   0.0041
## p7    0.0118   0.0119   0.0120   0.0110   0.0119   0.0120   0.0120
## p8    0.0127   0.0127   0.0126   0.0128   0.0126   0.0126   0.0126
## p9    0.0197   0.0197   0.0194   0.0209   0.0196   0.0194   0.0194
## p10   2.9572   2.9468   2.8880   3.2112   2.9218   2.8878   2.8891
##     Choice 8 Choice 9 Choice 10
## p1    0.5379   0.5379    0.5379
## p2    0.4343   0.4343    0.4343
## p3    0.0055   0.0055    0.0052
## p4    2.5264   2.5268    2.7038
## p5    0.2179   0.2179    0.1928
## p6    0.0041   0.0041    0.0033
## p7    0.0120   0.0120    0.0108
## p8    0.0126   0.0126    0.0128
## p9    0.0194   0.0194    0.0214
## p10   2.8893   2.8904    3.4126
```

Interpretation: the matrix above represents marginal effect of price on the probability of a family choosing product corresponding to choice i. The diagonal value of the matrix is the marginal effect from its own price, for example, 0.54 represents the fact that if price of product 1 “Pk_Stk” increases by $1 the probability of purchasing product 1 decreases by 0.5379 (decreases because the estimated coefficient is minus). The off-diagonal value of the matrix indicates marginal effect from its competing product price.

The second model (Multinomial Logit)

- Create individual likelihood function

```
expXB <- exp(X %*% mlogit_coeff)
pmlogit <- t(apply(expXB, 1, function(x) x / sum(x)))
```

- Select only family income to calculate

$$\beta_j$$

```
beta_fam <-  rowrep(mlogit_coeff[2,], nrow(pmlogit))
```

- Calculate

$$\bar{\beta}_i$$

```
beta_bar = pmlogit %*% t(mlogit_coeff)

## Write the column repclication function
colrep<-function(x,n){
  matrix(rep(x,each=n), ncol=n, byrow=TRUE)
}

## Replicate the column to make matrix conform
beta_bar <- beta_bar[,2] %>% colrep(10)
```

- Calculate

$$\beta_j - \bar{\beta}_i$$

```
beta_dif <- beta_fam - beta_bar
```

- Calculate the mean of

$$p_{ij}(\beta_j - \bar{\beta}_i)$$

  as the marginal effects for family income

```
beta_dif <- pmlogit*beta_dif
mfx_mlogit <- t(apply(beta_dif, 2, mean))
mfx_mlogit <-round(mfx_mlogit, digits = 4)
mfx_mlogit
```

```
##      Choice 1 Choice 2 Choice 3 Choice 4 Choice 5 Choice 6 Choice 7
## [1,]  -0.0012   -8e-04    9e-04    1e-04   -7e-04    4e-04   -8e-04
##      Choice 8 Choice 9 Choice 10
## [1,]   0.0011   0.0011    -1e-04
```

Interpretation: for choice 1, a increase in family income by $1,000 decreases the probability that the household purchases product of choice 1 by 0.0012

Exercise 5 Mixed Logit and IIA

- Write the likelihood function similar to Exercise 2 but with full matrix X and gamma

```
llxlogit <- function(coeff) {
  beta <- coeff[1]
  gamma <- matrix(coeff[2:length(coeff)], ncol = ncol(Z), byrow = TRUE)
  gamma[,1] <- 0
```

```
  ZB <- Z*beta
  XG <- X%*%gamma
  expZBXG <- exp(ZB+XG)
  pxlogit <- t(apply(expZBXG, 1, function(x) x / sum(x)))
  pxlogit <- pxlogit*D
  ll <- sum(log(rowSums(pxlogit)))
  return(-ll)
}
```

- Estimate the full mixed-logit model Note that there are 71 coefficients where 70 (gamma) are the multinomial part of the model (10 choices X 7 Characters) with all 7 gamma corresponding to choice 1 restricted to 0 PLUS one beta from the conditional part of the model corresponding prices.

```
G <- matrix(0,71)
xlogit <- nlm(llxlogit,G)
```

- Recover log-likelihood for the full model

```
ll_f <- -llxlogit(xlogit$estimate)
ll_f
```

```
## [1] -7131.247
```

- Create data set that does not include one choice here I omit choice 10 to estimate the alternative model

```
dfmar_alter <- dfmar[dfmar$choice != 10,]
dfmar_alter <- subset(dfmar_alter, select = -c(dfmar_alter$PPk_Stk))
Z <- as.matrix(dfmar_alter[,2:10])
X <- as.matrix(cbind(matrix(1,nrow(dfmar_alter)), dfmar_alter$Income,
                     dfmar_alter$Fs3_4, dfmar_alter$Fs5.,
                     dfmar_alter$college, dfmar_alter$whtcollar,
                     dfmar_alter$retired))
D <- dummy("choice", data = dfmar_alter, sep="")
```

- Estimate the alternative mixed-logit model Note that there are 64 coefficients where 63 (gamma) are the multinomial part of the model (10-1 choices X 7 Characters) with all 7 gamma corresponding to choice 1 restricted to 0 PLUS one beta from the conditional part of the model corresponding prices.

```
G_alt <- matrix(0,64)
xlogit_r <- nlm(llxlogit,G_alt)
```

- Recover log-likelihood for the alternative model

```
ll_r <- -llxlogit(xlogit_r$estimate)
ll_r
```

```
## [1] -6962.167
```

- Test for IIA Test for IIA using LR ratio

```
MTT <- -2*(ll_f-ll_r)
p_value <- pchisq(MTT, df=64, lower.tail = FALSE)
p_value
```

```
## [1] 6.495623e-39
```

Now that I eliminate choice 10, the largest market share, p-value is close to zero, that is IIA is violated.