> # Jeff Hill, ECON 613 Assignment 3

> library(bayesm)

> library(plyr)

> library(dplyr)

Attaching package: ‘dplyr’

The following objects are masked from ‘package:plyr’:

arrange, count, desc, failwith, id, mutate, rename, summarise, summarize

The following objects are masked from ‘package:stats’:

filter, lag

The following objects are masked from ‘package:base’:

intersect, setdiff, setequal, union

>

> # load data

> datapath <- "/Users/admin/Documents/Econ\_613/Data/Assignment 3/"

> demos <- read.csv(file=paste(datapath,"demos.csv",sep=""), header=TRUE, sep=",")

> product <- read.csv(file=paste(datapath,"product.csv",sep=""), header=TRUE, sep=",")

>

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> # Exercise 1 Data Description

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> # average and variance of each product:

> # mean

> prodmean <- apply(product[,4:13], 2, mean)

> prodmean

PPk\_Stk PBB\_Stk PFl\_Stk PHse\_Stk PGen\_Stk PImp\_Stk PSS\_Tub PPk\_Tub PFl\_Tub PHse\_Tub

0.5184362 0.5432103 1.0150201 0.4371477 0.3452819 0.7807785 0.8250895 1.0774094 1.1893758 0.5686734

> # standard deviation

> prodsd <- apply(product[,4:13], 2, sd)

> prodsd

PPk\_Stk PBB\_Stk PFl\_Stk PHse\_Stk PGen\_Stk PImp\_Stk PSS\_Tub PPk\_Tub PFl\_Tub PHse\_Tub

0.15051740 0.12033186 0.04289519 0.11883123 0.03516605 0.11464607 0.06121159 0.02972613 0.01405451 0.07245500

>

> # market share by product

> mshare <-table(product$choice)/(sum(table(product$choice)))

> mshare

1 2 3 4 5 6 7 8 9 10

0.39507830 0.15637584 0.05436242 0.13266219 0.07046980 0.01655481 0.07136465 0.04541387 0.05033557 0.00738255

>

> # market share by brand

> # first combine stick and tub for brands: PPk, PFl, and PHse as they are the only brands offering both stick and tub.

> brand\_choice <- product$choice

> brand\_choice[brand\_choice == 8] <- 1

> brand\_choice[brand\_choice == 9] <- 3

> brand\_choice[brand\_choice == 10] <- 4

> # now choices are grouped only by brand in brand\_choice, so again use table

> brand\_share <-table(brand\_choice)/(sum(table(brand\_choice)))

> brand\_share

brand\_choice

1 2 3 4 5 6 7

0.44049217 0.15637584 0.10469799 0.14004474 0.07046980 0.01655481 0.07136465

>

> # market share by observed characteristic, i.e. income level, family size, colllege, etc.

> # need to link demographic data to product data, so will create new df, and merge via hhid.

> df <- data.frame(merge(product, demos, by="hhid"))

> df$intercept <- 1 # add a column to df, containing all 1's for an intercept. this will be used in exercise 2 and 3.

>

> # create a vector of income levels

> inc\_levels <- c(seq(2.5, 47.5, by=5), 55, 67.5, 87.5, 130)

>

> # create an empty dataframe to insert income share data into.

> inc\_share\_df <- data.frame(prod1 = NA, prod2 = NA, prod3 = NA, prod4 = NA, prod5 = NA, prod6 = NA, prod7 = NA, prod8 = NA, prod9 = NA, prod10 = NA)

> colnames(inc\_share\_df) <- c(1:10) # rename column for sake of convenience, they will be renamed at the end.

>

> # here we use a nested for loop, to cycle through first income levels and then cycle through the 10 products inside that.

> # we insert the market share of each product at a given income level into a 14 x 10 dataframe.

> for(i in inc\_levels) {

+ for(j in c(1:10)) {

+ inc\_share\_df[i,j] <- sum(df[df["Income"] == i,'choice']==j)/length(df[df["Income"] == i,'choice'])

+ }

+ }

>

> # our nested for loop generated a bunch of empty rows because we indexed on income level which goes up to 130,

> # so we just drop the empty rows here.

> inc\_share\_df <- inc\_share\_df[rowSums(is.na(inc\_share\_df)) != ncol(inc\_share\_df),]

>

> # rename rows and columns

> rownames(inc\_share\_df) <- c('inc2.5','inc7.5','inc12.5','inc17.5','inc22.5','inc27.5','inc32.5',

+ 'inc37.5','inc42.5','inc47.5','inc55','inc67.5','inc87.5','inc130')

> colnames(inc\_share\_df) <- c('product1','product2','product3','product4','product5',

+ 'product6','product7','product8','product9','product10')

> inc\_share\_df # this is the final dataframe, containing product share broken down at each income level. for certain income levels,

product1 product2 product3 product4 product5 product6 product7 product8 product9

inc2.5 0.3800000 0.08000000 0.00000000 0.04000000 0.12000000 0.000000000 0.32000000 0.02000000 0.04000000

inc7.5 0.3966102 0.18305085 0.04406780 0.11525424 0.06440678 0.006779661 0.09152542 0.02033898 0.07457627

inc12.5 0.3959596 0.21414141 0.08282828 0.08888889 0.04646465 0.018181818 0.08080808 0.01616162 0.05050505

inc17.5 0.4697194 0.14771049 0.03988183 0.16395864 0.03101920 0.007385524 0.07976366 0.02806499 0.02954210

inc22.5 0.3463820 0.14590747 0.04033215 0.18268090 0.14590747 0.002372479 0.04863582 0.04270463 0.03558719

inc27.5 0.4096639 0.19747899 0.01890756 0.14075630 0.03781513 0.012605042 0.05042017 0.05252101 0.07142857

inc32.5 0.3806922 0.15300546 0.05100182 0.11657559 0.09836066 0.007285974 0.08925319 0.03460838 0.06010929

inc37.5 0.4731183 0.12186380 0.06093190 0.10394265 0.08243728 0.003584229 0.05376344 0.05017921 0.03225806

inc42.5 0.4125413 0.10891089 0.10891089 0.07590759 0.01980198 0.066006601 0.08910891 0.06930693 0.04620462

inc47.5 0.4414894 0.11702128 0.12234043 0.08510638 0.03723404 0.090425532 0.03191489 0.04787234 0.01063830

inc55 0.2338308 0.14925373 0.05472637 0.15920398 0.03482587 0.014925373 0.05970149 0.20895522 0.08457711

inc67.5 0.3725490 0.07843137 0.01960784 0.15686275 0.11764706 0.039215686 0.13725490 0.05882353 0.00000000

inc87.5 0.2432432 0.27027027 0.08108108 0.02702703 0.00000000 0.027027027 0.02702703 0.00000000 0.32432432

inc130 0.1923077 0.03846154 0.11538462 0.30769231 0.07692308 0.076923077 0.00000000 0.00000000 0.19230769

product10

inc2.5 0.000000000

inc7.5 0.003389831

inc12.5 0.006060606

inc17.5 0.002954210

inc22.5 0.009489917

inc27.5 0.008403361

inc32.5 0.009107468

inc37.5 0.017921147

inc42.5 0.003300330

inc47.5 0.015957447

inc55 0.000000000

inc67.5 0.019607843

inc87.5 0.000000000

inc130 0.000000000

> # the number of individuals was low, so there are some 0% market shares, for example at income level = 2.5 no one

> # bought product 3, 6, or 10.

>

>

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> # Exercise 2 First Model

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> # here we create the conditional logit negative log likelihood function. The proposed model specification includes all 10 prices for

> # the 10 products.

> clogit\_nll <- function (clogit\_parameters,x,y) { # here clogit\_parameters will be a 10x1 vector containing 1 beta(price)

+ # and 9 alphas, since we have 10 product choices and we force the first alpha

+ # to be zero.

+ clogit\_beta <- clogit\_parameters[1] # separating out beta and alphas from clogit\_parameters

+ alpha <- clogit\_parameters[2:10]

+ a\_mat <- cbind(0,matrix(alpha,nrow=4470,ncol=9,byrow=T)) # create alpha matrix to account for alternative specific constants.

+ # we cbind in a column of zeros to set the first alpha to our reference

+ xb <- x\*clogit\_beta + a\_mat # create XB, size:n x 10. Beta is just the single beta for price, and there are 9 alphas

+

+ rs <- rowSums(exp(xb)) # this is the denominator of P\_ij

+ xb\_vec <- rep(0, length(y)) # create an empty vector from which we will select the correct xb value,

+ # based on which choice individual i made.

+ for (i in 1:length(y)) { # this for loop goes through the matrix xb by row, and pulls out the correct xb value based on what choice

+ xb\_vec[i] <- xb[i,y[i]] # that individual made, and stores the value in xb\_vec. e^(xb\_vec) is now the numerator of P\_ij

+ }

+ -sum(xb\_vec - log(rs)) # here we calculate the negative log likelihood.

+ }

>

> cond\_x <- as.matrix(df[,c(4:13)]) # here we subset the x for the conditional logit out of df. we only pull out the columns of

> # interest, namely the price columns for the 10 products.

>

> #clogit\_estimates contains the optimized values for Beta, and the 9 alphas. note these are alphas for products 2 through 10,

> # as alpha for product 1 was bound = 0 to set it as our reference point.

> clogit\_estimates <- nlm(clogit\_nll,rep(0,10),x=cond\_x,y=df$choice)$estimate

> clogit\_price\_beta <-clogit\_estimates[1]

> clogit\_alphas <- clogit\_estimates[c(2:10)]

> clogit\_price\_beta # this is the beta for price

[1] -6.656581

> clogit\_alphas # these are the alphas for products 2 thorugh 10, relative to product 1 (alpha 1 = 0)

[1] -0.9543064 1.2969755 -1.7173331 -2.9040054 -1.5153134 0.2517577 1.4648557 2.3575077 -3.8965953

>

> #interpret the coefficient on price:

> # an increase in price of a product results in decreasing the probability of choosing that product

>

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> # Exercise 3 Second Model

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

>

> # here we create the multinomial logit log likelihood function. Much of the function is identical to the clogit\_nll function above.

> # our model specification includes income, the two family size dummy variables (with family size = 1 or 2 being the reference point),

> # college, whtcollar and retired.

> mlogit\_nll <- function (mlogit\_beta,x,y) {

+ mat <- matrix(mlogit\_beta,nrow=7,ncol=9, byrow=T) # turn out mlogit\_beta vector into a matrix of the correct size.

+ mat <- cbind(0,mat) # add on a column of zeros on front, to account for product 1 being the reference point.

+ xb <- x%\*%mat # create XB size:n x 10. We do this through matrix multiplication of x\_i \* Beta\_j, which differs from cond. logit.

+ rs <- rowSums(exp(xb)) # this is the denominator of multinomial logit probability P\_ij

+ xb\_vec <- rep(0, length(y))

+ for (i in 1:length(y)) { # again this for loop pulls out the correct xb term, based on the choice of that individual.

+ xb\_vec[i] <- xb[i,y[i]]

+ }

+ -sum(xb\_vec - log(rs)) # sum the log of each individual likelihood, then make negative

+ }

>

> multi\_x <- as.matrix(df[,c(22,15,16,17,19,20,21)]) # here subset x out of df, pulling out only the columns of interest: intercept,

> # income, Fs3\_4, Fs5.,whtcollar,retired, and college

>

> mlogit\_opt <- nlm(mlogit\_nll,rep(0,63),x=multi\_x,y=df$choice)

Warning messages:

1: In nlm(mlogit\_nll, rep(0, 63), x = multi\_x, y = df$choice) :

NA/Inf replaced by maximum positive value

2: In nlm(mlogit\_nll, rep(0, 63), x = multi\_x, y = df$choice) :

NA/Inf replaced by maximum positive value

>

> #mlogit\_estimates contains the optimized values for our 63 Betas. there are 9 for each independent variable, and 9 for the intercept.

> mlogit\_estimates <- mlogit\_opt$estimate

> m\_ind\_betas <- as.data.frame(matrix(mlogit\_estimates,nrow=7,ncol=9,byrow=T))

> rownames(m\_ind\_betas) <- c('intercept','income','Fs3\_4','Fs5.','college','whtcollar','retired')

> colnames(m\_ind\_betas) <- c('product2','product3','product4','product5',

+ 'product6','product7','product8','product9','product10')

> m\_ind\_betas # m\_ind\_betas contains the betas for the 6 individual variables, for products 2 thorugh 10 relative to product 1.

product2 product3 product4 product5 product6 product7 product8 product9

intercept -0.832729362 -3.20972954 -1.048176617 -2.35353303 -3.52598657 -0.830734109 -2.09735515 -2.32976170

income -0.002491885 0.02330197 0.003348327 -0.00763317 0.03004953 -0.007258172 0.02703394 0.02654402

Fs3\_4 -0.014046383 -1.00088707 -0.057175269 0.60737529 -1.42965016 -0.614731227 -0.41020337 -1.32849184

Fs5. -0.263049489 -1.20401750 0.344733939 0.83299119 0.27453471 -1.520966715 -1.40953455 -1.94493204

college 0.042842465 0.52185503 -0.250660076 -0.35583028 0.24559004 0.076652889 -0.40241269 -0.33596727

whtcollar -0.033414963 0.58916151 -0.010989053 0.66777348 -0.52172965 -0.100071733 -0.31961975 0.37034845

retired 0.122659803 1.46192648 -0.352307807 0.14342782 0.30917167 -1.073547320 -1.33852823 0.43123808

product10

intercept -3.672774096

income -0.006352414

Fs3\_4 -0.912077372

Fs5. 1.033344090

college 0.122874456

whtcollar 0.197825886

retired -1.210626769

>

> #interpret the coefficient on family: I was not sure if I was supposed to interpret the coefficient on

> # family (Fs3\_4,Fs5.) or family income, so I do both.

>

> # interpretation of income:

> m\_ind\_betas[2,]

product2 product3 product4 product5 product6 product7 product8 product9

income -0.002491885 0.02330197 0.003348327 -0.00763317 0.03004953 -0.007258172 0.02703394 0.02654402

product10

income -0.006352414

> # the coefficient from income for product 2 says an increase in income results in having a lower probability

> # of choosing product 2 relative to product 1. This is the same interpretation for all income coefficients

> # based on the SIGN of the coefficient (negative or positive). (This is also assuming significance)

>

> # I included the dummy variables for Family size 3-4 and family size >5

> m\_ind\_betas[3:4,] # these are the coefficients for the family dummys.

product2 product3 product4 product5 product6 product7 product8 product9 product10

Fs3\_4 -0.01404638 -1.000887 -0.05717527 0.6073753 -1.4296502 -0.6147312 -0.4102034 -1.328492 -0.9120774

Fs5. -0.26304949 -1.204018 0.34473394 0.8329912 0.2745347 -1.5209667 -1.4095345 -1.944932 1.0333441

> # the coefficient for Fs3\_4 for Product 2 says Having a family size of 3-4 compared to having a family size of 1-2

> # has a negative impact on your likelihood of choosing product 2 relative to product 1. (i.e you are more likely

> # to switch product 1 from product 2 if you "switch" from family size 1-2 to size 3-4, holding all else constant.

> # this holds the same for "switching" from family size 1-2 to size >5 based on the SIGN of the coefficient

> # (This is assuming significance)

>

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> # Exercise 4 Marginal Effects

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

>

> #NOTE: I did not complete this section before the midnight deadline, I only made it through part of the

> # multinomial logit marginal effects.

>

> mlogit\_p <- function (mlogit\_beta,x,y) {

+ mat <- matrix(mlogit\_beta,nrow=7,ncol=9, byrow=T)

+ mat <- cbind(0,mat)

+ xb <- x%\*%mat

+ rs <- rowSums(exp(xb))

+ exp(xb)/rs

+ }

> p <- mlogit\_p(mlogit\_estimates,multi\_x,df$choice) #following the formula for multinomial

> # marginal effects, here we calculate the matrix p containing all p\_ij

> # the p matrix must then be multiplied by (B\_j - B\_i\_bar), where B\_i\_bar = the sum of p\_il\*B\_l

> me\_mlogit\_betas <- t(cbind(0,m\_ind\_betas))

> B\_bar <- p%\*%me\_mlogit\_betas

>

>

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> # Exercise 5 IIA

> #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> # here we construct the mixed logit negative log likelihood function, which combines product and individual characteristics.

> mixedlogit\_nll <- function (mixedlogit\_parameters,x,y) { # mixedlogit\_parameters is now a vector of length 64, 1 price beta and

+ # 63 individual characteristic betas.

+ c\_x <- x[,c(1:10)] # here since x will now contain product data AND demographic data, we separate them into

+ m\_x <- x[,c(11:17)] # c\_x containing product data and m\_x containing demographic data.

+ beta <- mixedlogit\_parameters[1] # separating out beta, and gammas from mixedlogit\_parameters

+ gamma <- mixedlogit\_parameters[2:64]

+ # there is no need for an alpha matrix in this specification, since the intercept is already included in m\_x

+ # we cbind in a column of zeros to set the first alpha to our reference

+ g\_mat <- matrix(gamma,nrow=7,ncol=9, byrow=T) #create gamma matrix for demographic data

+ g\_mat <- cbind(0,g\_mat)

+ xb <- c\_x\*beta + m\_x%\*%g\_mat # create XB now with c\_x and m\_x

+ rs <- rowSums(exp(xb)) # this is the denominator of P\_ij

+ xb\_vec <- rep(0, length(y)) # create an empty vector from which we will select the correct xb value,

+ # based on which choice individual i made.

+ for (i in 1:length(y)) { # this for loop goes through the matrix xb by row, and pulls out the correct xb value based on what choice

+ xb\_vec[i] <- xb[i,y[i]] # that individual made, and stores the value in xb\_vec. e^(xb\_vec) is now the numerator of P\_ij

+ }

+ -sum(xb\_vec - log(rs)) # here we calculate the negative log likelihood.

+ }

>

> mixed\_x <- as.matrix(df[,c(4:13,22,15,16,17,19,20,21)]) #this is the subset of df, containing all relevant x data.

>

> mixed\_logit\_opt <- nlm(mixedlogit\_nll,rep(0,64),x=mixed\_x,y=df$choice) #the optimization

Warning messages:

1: In nlm(mixedlogit\_nll, rep(0, 64), x = mixed\_x, y = df$choice) :

NA/Inf replaced by maximum positive value

2: In nlm(mixedlogit\_nll, rep(0, 64), x = mixed\_x, y = df$choice) :

NA/Inf replaced by maximum positive value

> mixed\_logit\_estimates <- mixed\_logit\_opt$estimate

> mixed\_price\_beta <- mixed\_logit\_estimates[1]

> mixed\_price\_beta # this is the beta for price in our mixed model.

[1] -6.803308

> mixed\_ind\_betas <- as.data.frame(matrix(mixed\_logit\_estimates[2:64],nrow=7,ncol=9,byrow=T))

> rownames(mixed\_ind\_betas) <- c('intercept','income','Fs3\_4','Fs5.','college','whtcollar','retired')

> colnames(mixed\_ind\_betas) <- c('product2','product3','product4','product5',

+ 'product6','product7','product8','product9','product10')

> mixed\_ind\_betas # mixed\_ind\_betas contains the betas for the 6 individual variables, for products 2 thorugh 10 relative to product 1

product2 product3 product4 product5 product6 product7 product8 product9

intercept -1.0150465535 -0.1127491 -1.876221932 -3.601468194 -2.09239777 1.069930273 1.34905037 1.91655687

income -0.0007407996 0.0226821 0.002461065 -0.007941926 0.02814744 -0.005951668 0.02795926 0.02649195

Fs3\_4 0.0592045172 -0.8502664 0.120619653 0.610130172 -1.17897201 -0.578413133 -0.24370468 -1.19177107

Fs5. -0.2075018861 -0.9295348 0.483779241 0.848454090 0.45789573 -1.398381046 -0.97033015 -1.54851639

college 0.0453015761 0.5565059 -0.278178773 -0.358549127 0.24828010 0.060741166 -0.36066116 -0.35352583

whtcollar -0.0281834865 0.6789456 0.082405523 0.676305155 -0.32010056 -0.129945736 -0.38101504 0.48800632

retired 0.2508652170 1.8845894 -0.087465611 0.436756449 0.79090726 -0.873794723 -0.62282862 0.84391630

product10

intercept -3.475326803

income -0.009506217

Fs3\_4 -0.914244592

Fs5. 0.970797798

college 0.194931322

whtcollar 0.144977619

retired -1.052201751

> # from the mixed logit regression. This is Beta^f in the assignment.

>

> # Now an alternative specification: I will remove data for choice 4. This involves removing associated column, but also all

> # observations that selected choice 4.

> df\_cut <- df[df$choice!=4,] # removed all observations where someone chose product 4

> df\_cut <- subset(df\_cut, select=-c(PHse\_Stk))

> # now since 4 has been removed, I will shift choice 5 to 4, 6 to 5, etc.

> for (i in 1:3877) { # this quick forloop finds all values of choice above 4, and reduces them by 1.

+ if (df\_cut[['choice']][i] > 4) {

+ df\_cut[['choice']][i] <- df\_cut[['choice']][i]-1

+ }

+ }

>

> # now modify the mixed logit log likelihood function to run on the reduced dataset:

> cut\_mixedlogit\_nll <- function (mixedlogit\_parameters,x,y) { # It is exactly the same, with a couple vector lengths shifted.

+ # 56 individual characteristic betas. (7x8 now)

+ c\_x <- x[,c(1:9)]

+ m\_x <- x[,c(10:16)]

+ beta <- mixedlogit\_parameters[1]

+ gamma <- mixedlogit\_parameters[2:57]

+ g\_mat <- matrix(gamma,nrow=7,ncol=8, byrow=T)

+ g\_mat <- cbind(0,g\_mat)

+ xb <- c\_x\*beta + m\_x%\*%g\_mat

+ rs <- rowSums(exp(xb))

+ xb\_vec <- rep(0, length(y))

+ for (i in 1:length(y)) {

+ xb\_vec[i] <- xb[i,y[i]]

+ }

+ -sum(xb\_vec - log(rs))

+ }

>

> cut\_x <- as.matrix(df\_cut[,c(4:12,21,14,15,16,18,19,20)])

>

> cut\_mixed\_logit\_opt <- nlm(cut\_mixedlogit\_nll,rep(0,57),x=cut\_x,y=df\_cut$choice) #the optimization

Warning messages:

1: In nlm(cut\_mixedlogit\_nll, rep(0, 57), x = cut\_x, y = df\_cut$choice) :

NA/Inf replaced by maximum positive value

2: In nlm(cut\_mixedlogit\_nll, rep(0, 57), x = cut\_x, y = df\_cut$choice) :

NA/Inf replaced by maximum positive value

> cut\_mixed\_logit\_estimates <- cut\_mixed\_logit\_opt$estimate

> cut\_mixed\_price\_beta <- cut\_mixed\_logit\_estimates[1]

> cut\_mixed\_price\_beta # this is the beta for price in our mixed model.

[1] -6.494169

> cut\_mixed\_ind\_betas <- as.data.frame(matrix(cut\_mixed\_logit\_estimates[2:57],nrow=7,ncol=8,byrow=T))

> rownames(cut\_mixed\_ind\_betas) <- c('intercept','income','Fs3\_4','Fs5.','college','whtcollar','retired')

> colnames(cut\_mixed\_ind\_betas) <- c('product2','product3','product5',

+ 'product6','product7','product8','product9','product10')

> cut\_mixed\_ind\_betas # mixed\_ind\_betas contains the betas for the 6 individual variables, for products 2 through 10 (minus 4 now)

product2 product3 product5 product6 product7 product8 product9 product10

intercept -0.998135027 -0.41720324 -3.556787377 -2.28272514 0.951642726 1.18580035 1.64348046 -3.533687099

income -0.001619728 0.02571825 -0.006672814 0.03286316 -0.004108841 0.02903965 0.02743131 -0.007732867

Fs3\_4 0.081496437 -0.85920239 0.632732182 -1.24840184 -0.592214406 -0.28192209 -1.19040750 -0.891803786

Fs5. -0.191596863 -0.94533035 0.862082547 0.53065813 -1.421668390 -0.91162228 -1.55017444 1.032671035

college 0.048676256 0.53032083 -0.366177480 0.27754510 0.034003626 -0.35363660 -0.34606607 0.190497211

whtcollar -0.046382280 0.77762730 0.628494986 -0.40590989 -0.144441236 -0.38025778 0.56664084 0.116511226

retired 0.322746624 2.00785932 0.495640142 0.84006222 -0.900015954 -0.61991010 0.92760667 -1.110193815

> # relative to product 1 from the mixed logit regression.

>

> # here we calculate the negative log likelihood of the restricted model (less product 4) of the restricted betas.

> restricted\_lik <- cut\_mixedlogit\_nll(cut\_mixed\_logit\_estimates,cut\_x,df\_cut$choice)

>

> # to calculate the same for the unrestricted betas (beta^f), we need to select the proper betas out of our beta vector

> # selecting all betas except those associated with product 4:

> unrestricted\_betas <- mixed\_logit\_estimates[-c(4,13,22,31,40,49,58)]

> unrestricted\_lik <- cut\_mixedlogit\_nll(unrestricted\_betas,cut\_x,df\_cut$choice)

>

> # construct test statistic:

> MTT <- 2\*(unrestricted\_lik-restricted\_lik) #since our log likelihood function is a negative log likelihood function,

> # multiplying by another negative negates the minus in front.

> MTT

[1] 2.960407

> p

[,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]

[1,] 0.33916312 0.16049581 0.212267482 0.07254139 0.020337524 0.046148572 0.043075829 0.017580460

[2,] 0.33916312 0.16049581 0.212267482 0.07254139 0.020337524 0.046148572 0.043075829 0.017580460

[3,] 0.33916312 0.16049581 0.212267482 0.07254139 0.020337524 0.046148572 0.043075829 0.017580460

[4,] 0.33916312 0.16049581 0.212267482 0.07254139 0.020337524 0.046148572 0.043075829 0.017580460

[5,] 0.33916312 0.16049581 0.212267482 0.07254139 0.020337524 0.046148572 0.043075829 0.017580460

[6,] 0.33916312 0.16049581 0.212267482 0.07254139 0.020337524 0.046148572 0.043075829 0.017580460

[7,] 0.33916312 0.16049581 0.212267482 0.07254139 0.020337524 0.046148572 0.043075829 0.017580460

[8,] 0.42394281 0.19860187 0.123929081 0.08055059 0.102026377 0.005221897 0.029375149 0.007060905

[9,] 0.42394281 0.19860187 0.123929081 0.08055059 0.102026377 0.005221897 0.029375149 0.007060905

[10,] 0.42394281 0.19860187 0.123929081 0.08055059 0.102026377 0.005221897 0.029375149 0.007060905

[11,] 0.42394281 0.19860187 0.123929081 0.08055059 0.102026377 0.005221897 0.029375149 0.007060905

[12,] 0.42394281 0.19860187 0.123929081 0.08055059 0.102026377 0.005221897 0.029375149 0.007060905

[13,] 0.42394281 0.19860187 0.123929081 0.08055059 0.102026377 0.005221897 0.029375149 0.007060905

[14,] 0.42394281 0.19860187 0.123929081 0.08055059 0.102026377 0.005221897 0.029375149 0.007060905

[15,] 0.42394281 0.19860187 0.123929081 0.08055059 0.102026377 0.005221897 0.029375149 0.007060905

[16,] 0.42394281 0.19860187 0.123929081 0.08055059 0.102026377 0.005221897 0.029375149 0.007060905

[17,] 0.42394281 0.19860187 0.123929081 0.08055059 0.102026377 0.005221897 0.029375149 0.007060905

[18,] 0.42394281 0.19860187 0.123929081 0.08055059 0.102026377 0.005221897 0.029375149 0.007060905

[19,] 0.33738022 0.15106275 0.140784646 0.09428182 0.027795148 0.041731366 0.038273114 0.029937139

[20,] 0.33738022 0.15106275 0.140784646 0.09428182 0.027795148 0.041731366 0.038273114 0.029937139

[21,] 0.33738022 0.15106275 0.140784646 0.09428182 0.027795148 0.041731366 0.038273114 0.029937139

[22,] 0.33738022 0.15106275 0.140784646 0.09428182 0.027795148 0.041731366 0.038273114 0.029937139

[23,] 0.33738022 0.15106275 0.140784646 0.09428182 0.027795148 0.041731366 0.038273114 0.029937139

[24,] 0.33738022 0.15106275 0.140784646 0.09428182 0.027795148 0.041731366 0.038273114 0.029937139

[25,] 0.33738022 0.15106275 0.140784646 0.09428182 0.027795148 0.041731366 0.038273114 0.029937139

[26,] 0.33738022 0.15106275 0.140784646 0.09428182 0.027795148 0.041731366 0.038273114 0.029937139

[27,] 0.33738022 0.15106275 0.140784646 0.09428182 0.027795148 0.041731366 0.038273114 0.029937139

[28,] 0.33738022 0.15106275 0.140784646 0.09428182 0.027795148 0.041731366 0.038273114 0.029937139

[29,] 0.33738022 0.15106275 0.140784646 0.09428182 0.027795148 0.041731366 0.038273114 0.029937139

[30,] 0.35540351 0.14309396 0.038879622 0.13067115 0.057622538 0.010500306 0.123399856 0.050875059

[31,] 0.35540351 0.14309396 0.038879622 0.13067115 0.057622538 0.010500306 0.123399856 0.050875059

[32,] 0.20238394 0.07143763 0.190637932 0.07320908 0.013472820 0.062638132 0.045646345 0.128541067

[33,] 0.20238394 0.07143763 0.190637932 0.07320908 0.013472820 0.062638132 0.045646345 0.128541067

[34,] 0.39658503 0.18898589 0.092421724 0.10192733 0.039542286 0.023143054 0.053940517 0.017902338

[35,] 0.38015803 0.16518931 0.038880662 0.10998523 0.022145698 0.024192853 0.157509817 0.050094593

[36,] 0.38015803 0.16518931 0.038880662 0.10998523 0.022145698 0.024192853 0.157509817 0.050094593

[37,] 0.38015803 0.16518931 0.038880662 0.10998523 0.022145698 0.024192853 0.157509817 0.050094593

[38,] 0.38015803 0.16518931 0.038880662 0.10998523 0.022145698 0.024192853 0.157509817 0.050094593

[39,] 0.38015803 0.16518931 0.038880662 0.10998523 0.022145698 0.024192853 0.157509817 0.050094593

[40,] 0.38015803 0.16518931 0.038880662 0.10998523 0.022145698 0.024192853 0.157509817 0.050094593

[41,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[42,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[43,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[44,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[45,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[46,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[47,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[48,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[49,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[50,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[51,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[52,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[53,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[54,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[55,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[56,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[57,] 0.38676014 0.18202193 0.101269587 0.10108038 0.037118631 0.026228762 0.050729382 0.019985664

[58,] 0.42897740 0.13516903 0.029948461 0.17918010 0.103844380 0.028794225 0.032677437 0.013143343

[59,] 0.42897740 0.13516903 0.029948461 0.17918010 0.103844380 0.028794225 0.032677437 0.013143343

[60,] 0.42897740 0.13516903 0.029948461 0.17918010 0.103844380 0.028794225 0.032677437 0.013143343

[61,] 0.42897740 0.13516903 0.029948461 0.17918010 0.103844380 0.028794225 0.032677437 0.013143343

[62,] 0.42897740 0.13516903 0.029948461 0.17918010 0.103844380 0.028794225 0.032677437 0.013143343

[63,] 0.42897740 0.13516903 0.029948461 0.17918010 0.103844380 0.028794225 0.032677437 0.013143343

[64,] 0.42897740 0.13516903 0.029948461 0.17918010 0.103844380 0.028794225 0.032677437 0.013143343

[65,] 0.42897740 0.13516903 0.029948461 0.17918010 0.103844380 0.028794225 0.032677437 0.013143343

[66,] 0.42897740 0.13516903 0.029948461 0.17918010 0.103844380 0.028794225 0.032677437 0.013143343

[67,] 0.26278473 0.09749793 0.155322744 0.08890083 0.020379028 0.044591900 0.068528902 0.097196850

[68,] 0.26278473 0.09749793 0.155322744 0.08890083 0.020379028 0.044591900 0.068528902 0.097196850

[69,] 0.26278473 0.09749793 0.155322744 0.08890083 0.020379028 0.044591900 0.068528902 0.097196850

[70,] 0.26278473 0.09749793 0.155322744 0.08890083 0.020379028 0.044591900 0.068528902 0.097196850

[71,] 0.26278473 0.09749793 0.155322744 0.08890083 0.020379028 0.044591900 0.068528902 0.097196850

[72,] 0.39634195 0.11816681 0.018448921 0.21629998 0.131819158 0.024184346 0.026967008 0.020787895

[73,] 0.39634195 0.11816681 0.018448921 0.21629998 0.131819158 0.024184346 0.026967008 0.020787895

[74,] 0.39634195 0.11816681 0.018448921 0.21629998 0.131819158 0.024184346 0.026967008 0.020787895

[75,] 0.36352467 0.15087896 0.075296112 0.10577971 0.039746071 0.015956007 0.131418295 0.039834093

[76,] 0.36352467 0.15087896 0.075296112 0.10577971 0.039746071 0.015956007 0.131418295 0.039834093

[77,] 0.40535963 0.19558908 0.084077279 0.10245285 0.041989545 0.020355094 0.057171580 0.015984920

[78,] 0.40535963 0.19558908 0.084077279 0.10245285 0.041989545 0.020355094 0.057171580 0.015984920

[79,] 0.40535963 0.19558908 0.084077279 0.10245285 0.041989545 0.020355094 0.057171580 0.015984920

[80,] 0.40535963 0.19558908 0.084077279 0.10245285 0.041989545 0.020355094 0.057171580 0.015984920

[81,] 0.40535963 0.19558908 0.084077279 0.10245285 0.041989545 0.020355094 0.057171580 0.015984920

[82,] 0.40535963 0.19558908 0.084077279 0.10245285 0.041989545 0.020355094 0.057171580 0.015984920

[83,] 0.40535963 0.19558908 0.084077279 0.10245285 0.041989545 0.020355094 0.057171580 0.015984920

[84,] 0.40535963 0.19558908 0.084077279 0.10245285 0.041989545 0.020355094 0.057171580 0.015984920

[85,] 0.40535963 0.19558908 0.084077279 0.10245285 0.041989545 0.020355094 0.057171580 0.015984920

[86,] 0.25921158 0.10948588 0.246126760 0.07408527 0.038580128 0.025699017 0.024742645 0.021894895

[87,] 0.25921158 0.10948588 0.246126760 0.07408527 0.038580128 0.025699017 0.024742645 0.021894895

[88,] 0.25921158 0.10948588 0.246126760 0.07408527 0.038580128 0.025699017 0.024742645 0.021894895

[89,] 0.25921158 0.10948588 0.246126760 0.07408527 0.038580128 0.025699017 0.024742645 0.021894895

[90,] 0.25921158 0.10948588 0.246126760 0.07408527 0.038580128 0.025699017 0.024742645 0.021894895

[91,] 0.25921158 0.10948588 0.246126760 0.07408527 0.038580128 0.025699017 0.024742645 0.021894895

[92,] 0.25921158 0.10948588 0.246126760 0.07408527 0.038580128 0.025699017 0.024742645 0.021894895

[93,] 0.25921158 0.10948588 0.246126760 0.07408527 0.038580128 0.025699017 0.024742645 0.021894895

[94,] 0.25921158 0.10948588 0.246126760 0.07408527 0.038580128 0.025699017 0.024742645 0.021894895

[95,] 0.25921158 0.10948588 0.246126760 0.07408527 0.038580128 0.025699017 0.024742645 0.021894895

[96,] 0.43224522 0.17470835 0.036973343 0.12079212 0.083501646 0.005278167 0.081492350 0.035975400

[97,] 0.43224522 0.17470835 0.036973343 0.12079212 0.083501646 0.005278167 0.081492350 0.035975400

[98,] 0.43224522 0.17470835 0.036973343 0.12079212 0.083501646 0.005278167 0.081492350 0.035975400

[99,] 0.43224522 0.17470835 0.036973343 0.12079212 0.083501646 0.005278167 0.081492350 0.035975400

[100,] 0.43224522 0.17470835 0.036973343 0.12079212 0.083501646 0.005278167 0.081492350 0.035975400

[,9] [,10]

[1,] 0.086027830 0.0023619825

[2,] 0.086027830 0.0023619825

[3,] 0.086027830 0.0023619825

[4,] 0.086027830 0.0023619825

[5,] 0.086027830 0.0023619825

[6,] 0.086027830 0.0023619825

[7,] 0.086027830 0.0023619825

[8,] 0.027701445 0.0015898719

[9,] 0.027701445 0.0015898719

[10,] 0.027701445 0.0015898719

[11,] 0.027701445 0.0015898719

[12,] 0.027701445 0.0015898719

[13,] 0.027701445 0.0015898719

[14,] 0.027701445 0.0015898719

[15,] 0.027701445 0.0015898719

[16,] 0.027701445 0.0015898719

[17,] 0.027701445 0.0015898719

[18,] 0.027701445 0.0015898719

[19,] 0.136740857 0.0020129356

[20,] 0.136740857 0.0020129356

[21,] 0.136740857 0.0020129356

[22,] 0.136740857 0.0020129356

[23,] 0.136740857 0.0020129356

[24,] 0.136740857 0.0020129356

[25,] 0.136740857 0.0020129356

[26,] 0.136740857 0.0020129356

[27,] 0.136740857 0.0020129356

[28,] 0.136740857 0.0020129356

[29,] 0.136740857 0.0020129356

[30,] 0.079707274 0.0098467195

[31,] 0.079707274 0.0098467195

[32,] 0.207968693 0.0040643613

[33,] 0.207968693 0.0040643613

[34,] 0.082778393 0.0027734289

[35,] 0.042071828 0.0097719901

[36,] 0.042071828 0.0097719901

[37,] 0.042071828 0.0097719901

[38,] 0.042071828 0.0097719901

[39,] 0.042071828 0.0097719901

[40,] 0.042071828 0.0097719901

[41,] 0.092185365 0.0026201632

[42,] 0.092185365 0.0026201632

[43,] 0.092185365 0.0026201632

[44,] 0.092185365 0.0026201632

[45,] 0.092185365 0.0026201632

[46,] 0.092185365 0.0026201632

[47,] 0.092185365 0.0026201632

[48,] 0.092185365 0.0026201632

[49,] 0.092185365 0.0026201632

[50,] 0.092185365 0.0026201632

[51,] 0.092185365 0.0026201632

[52,] 0.092185365 0.0026201632

[53,] 0.092185365 0.0026201632

[54,] 0.092185365 0.0026201632

[55,] 0.092185365 0.0026201632

[56,] 0.092185365 0.0026201632

[57,] 0.092185365 0.0026201632

[58,] 0.012820586 0.0354450356

[59,] 0.012820586 0.0354450356

[60,] 0.012820586 0.0354450356

[61,] 0.012820586 0.0354450356

[62,] 0.012820586 0.0354450356

[63,] 0.012820586 0.0354450356

[64,] 0.012820586 0.0354450356

[65,] 0.012820586 0.0354450356

[66,] 0.012820586 0.0354450356

[67,] 0.158804804 0.0059922890

[68,] 0.158804804 0.0059922890

[69,] 0.158804804 0.0059922890

[70,] 0.158804804 0.0059922890

[71,] 0.158804804 0.0059922890

[72,] 0.018927438 0.0280564847

[73,] 0.018927438 0.0280564847

[74,] 0.018927438 0.0280564847

[75,] 0.066533596 0.0110324849

[76,] 0.066533596 0.0110324849

[77,] 0.074093748 0.0029262761

[78,] 0.074093748 0.0029262761

[79,] 0.074093748 0.0029262761

[80,] 0.074093748 0.0029262761

[81,] 0.074093748 0.0029262761

[82,] 0.074093748 0.0029262761

[83,] 0.074093748 0.0029262761

[84,] 0.074093748 0.0029262761

[85,] 0.074093748 0.0029262761

[86,] 0.198404977 0.0017688507

[87,] 0.198404977 0.0017688507

[88,] 0.198404977 0.0017688507

[89,] 0.198404977 0.0017688507

[90,] 0.198404977 0.0017688507

[91,] 0.198404977 0.0017688507

[92,] 0.198404977 0.0017688507

[93,] 0.198404977 0.0017688507

[94,] 0.198404977 0.0017688507

[95,] 0.198404977 0.0017688507

[96,] 0.023928762 0.0051046432

[97,] 0.023928762 0.0051046432

[98,] 0.023928762 0.0051046432

[99,] 0.023928762 0.0051046432

[100,] 0.023928762 0.0051046432

[ reached getOption("max.print") -- omitted 4370 rows ]

> qchisq(.001,df= 57)

[1] 29.59177

> # since our test statistic MTT is less than the critical value for the chi- squared distribution at alpha = .95

> # we conclude that IIA does hold. IIA may not hold depending on the product choice. Here product 4 was chosen roughly

> # 500 times out of the 4470 choices, but selecting a more popular choice to remove from the data might have a

> # more powerful effect on altering IAA.