R Code

# main code file

setwd("C:/z\_toshiba/course work/phd/econ 761/hw/hw4/")

rm(list=ls())

library(readxl)

library(tidyverse)

library(MASS)

library(expm)

library(Matrix)

library(dummies)

library(plyr)

library(FixedPoint)

source("blp.R")

source("blpmerger.R")

# import data

x <- read\_excel("cereal\_ps3.xls")

d <- read\_excel("demog\_ps3.xls")

x$t <- paste(x$city, x$quarter, sep="\_")

d$t <- paste(d$city, d$quarter, sep="\_")

x$brand <- paste(x$firm, x$brand, sep="\_")

x$brand <- as.factor(x$brand)

x$t <- as.factor(x$t)

x$id <- as.factor(x$id)

x$firm <- as.factor(x$firm)

x <- merge(x, d, by="t")

x$constant <- rep(1, nrow(x))

n <- 20

theta\_nlin=rep(0,20)

###########################################################

# OLS without brand FE (1-3)

blp(theta\_nlin=theta\_nlin, x=x, brandFE=F, iv=F, supply=F, n=n, A=1)

summary(mean\_u) # regression results

theta\_lin=coefficients(mean\_u)[1:4]

c(mean(b), quantile(b, 0.5), sd(b)) # markup statistics

c(mean(mc), quantile(mc, 0.5), sd(mc)) # marginal cost statistics

# Post-Nabisco merger

PNmerger <- x

PNmerger$xi <- residuals(mean\_u)

PNmerger$mc <- mc

PNmerger[which(PNmerger$firm==3), "firm"] <- 6

PNmerger$firm <- as.factor(as.character(PNmerger$firm))

mer <- blpmerger(theta\_nlin=theta\_nlin, theta\_lin=theta\_lin, x=PNmerger, va=va, n=n, brandFE=F)

c(mean(mer[[1]]), quantile(mer[[1]], 0.5), sd(mer[[1]]))

c(mean(mer[[2]]), quantile(mer[[2]], 0.5), sd(mer[[2]]))

#GM-Quaker merger

GQmerger <- x

GQmerger$xi <- residuals(mean\_u)

GQmerger$mc <- mc

GQmerger[which(GQmerger$firm==2), "firm"] <- 4

GQmerger$firm <- as.factor(as.character(GQmerger$firm))

mer <- blpmerger(theta\_nlin=theta\_nlin, theta\_lin=theta\_lin, x=GQmerger, va=va, n=n, brandFE=F)

c(mean(mer[[1]]), quantile(mer[[1]], 0.5), sd(mer[[1]]))

c(mean(mer[[2]]), quantile(mer[[2]], 0.5), sd(mer[[2]]))

###########################################################

###########################################################

# OLS with brand FE (1-3)

Z <- data.matrix(dummy.data.frame(as.data.frame(x$brand)))

blp(theta\_nlin=theta\_nlin, x=x, brandFE=T, iv=F, supply=F, n=n, A=(t(Z)%\*%Z))

coefs # regression results (coefficient estimates)

se # regression results (standard errors)

theta\_lin=coefs

c(mean(b), quantile(b, 0.5), sd(b)) # markup statistics

c(mean(mc), quantile(mc, 0.5), sd(mc)) # marginal cost statistics

nam <- names(coefficients(mean\_u)[2:length(coefficients(mean\_u))])

nam <- gsub("brand", "", nam)

resmd <- data.frame(nam, resmd)

colnames(resmd) <- c("brand", "resmd")

#Post-Nabisco merger

PNmerger <- x

PNmerger$xi <- residuals(mean\_u)

PNmerger$mc <- mc

PNmerger[which(PNmerger$firm==3), "firm"] <- 6

PNmerger$firm <- as.factor(as.character(PNmerger$firm))

PNmerger <- join(PNmerger, resmd, by="brand")

mer <- blpmerger(theta\_nlin=theta\_nlin, theta\_lin=theta\_lin, x=PNmerger, va=va, n=n, brandFE=T)

c(mean(mer[[1]]), quantile(mer[[1]], 0.5), sd(mer[[1]]))

c(mean(mer[[2]]), quantile(mer[[2]], 0.5), sd(mer[[2]]))

#GM-Quaker merger

GQmerger <- x

GQmerger$xi <- residuals(mean\_u)

GQmerger$mc <- mc

GQmerger[which(GQmerger$firm==2), "firm"] <- 4

GQmerger$firm <- as.factor(as.character(GQmerger$firm))

GQmerger <- join(GQmerger, resmd, by="brand")

mer <- blpmerger(theta\_nlin=theta\_nlin, theta\_lin=theta\_lin, x=GQmerger, va=va, n=n, brandFE=T)

c(mean(mer[[1]]), quantile(mer[[1]], 0.5), sd(mer[[1]]))

c(mean(mer[[2]]), quantile(mer[[2]], 0.5), sd(mer[[2]]))

###########################################################

###########################################################

# IV without brand FE (1-3)

vars <- seq(1, 20, 1)

vars <- paste("z", vars, sep="")

Z <- data.matrix(x[,vars])

blp(theta\_nlin=theta\_nlin, x=x, brandFE=F, iv=T, supply=F, n=n, A=(t(Z)%\*%Z))

summary(mean\_u) # regression results

theta\_lin=coefficients(mean\_u)

c(mean(b), quantile(b, 0.5), sd(b)) # markup statistics

c(mean(mc), quantile(mc, 0.5), sd(mc)) # marginal cost statistics

#Post-Nabisco merger

PNmerger <- x

PNmerger$xi <- residuals(mean\_u)

PNmerger$mc <- mc

PNmerger[which(PNmerger$firm==3), "firm"] <- 6

PNmerger$firm <- as.factor(as.character(PNmerger$firm))

mer <- blpmerger(theta\_nlin=theta\_nlin, theta\_lin=theta\_lin, x=PNmerger, va=va, n=n, brandFE=F)

c(mean(mer[[1]]), quantile(mer[[1]], 0.5), sd(mer[[1]]))

c(mean(mer[[2]]), quantile(mer[[2]], 0.5), sd(mer[[2]]))

#GM-Quaker merger

GQmerger <- x

GQmerger$xi <- residuals(mean\_u)

GQmerger$mc <- mc

GQmerger[which(GQmerger$firm==2), "firm"] <- 4

GQmerger$firm <- as.factor(as.character(GQmerger$firm))

mer <- blpmerger(theta\_nlin=theta\_nlin, theta\_lin=theta\_lin, x=GQmerger, va=va, n=n, brandFE=F)

c(mean(mer[[1]]), quantile(mer[[1]], 0.5), sd(mer[[1]]))

c(mean(mer[[2]]), quantile(mer[[2]], 0.5), sd(mer[[2]]))

###########################################################

###########################################################

# IV with brand FE (1-3)

vars <- seq(1, 20, 1)

vars <- paste("z", vars, sep="")

Z <- data.matrix(cbind(x[,vars], data.matrix(dummy.data.frame(as.data.frame(x$brand)))))

blp(theta\_nlin=theta\_nlin, x=x, brandFE=T, iv=T, supply=F, n=n, A=(t(Z)%\*%Z))

coefs # regression results (coefficient estimates)

se # regression results (standard errors)

theta\_lin=coefs

c(mean(b), quantile(b, 0.5), sd(b)) # markup statistics

c(mean(mc), quantile(mc, 0.5), sd(mc)) # marginal cost statistics

nam <- names(coefficients(mean\_u)[2:length(coefficients(mean\_u))])

nam <- gsub("brand", "", nam)

resmd <- data.frame(nam, resmd)

colnames(resmd) <- c("brand", "resmd")

#Post-Nabisco merger

PNmerger <- x

PNmerger$xi <- residuals(mean\_u)

PNmerger$mc <- mc

PNmerger[which(PNmerger$firm==3), "firm"] <- 6

PNmerger$firm <- as.factor(as.character(PNmerger$firm))

PNmerger <- join(PNmerger, resmd, by="brand")

mer <- blpmerger(theta\_nlin=theta\_nlin, theta\_lin=theta\_lin, x=PNmerger, va=va, n=n, brandFE=T)

c(mean(mer[[1]]), quantile(mer[[1]], 0.5), sd(mer[[1]]))

c(mean(mer[[2]]), quantile(mer[[2]], 0.5), sd(mer[[2]]))

#GM-Quaker merger

GQmerger <- x

GQmerger$xi <- residuals(mean\_u)

GQmerger$mc <- mc

GQmerger[which(GQmerger$firm==2), "firm"] <- 4

GQmerger$firm <- as.factor(as.character(GQmerger$firm))

GQmerger <- join(GQmerger, resmd, by="brand")

mer <- blpmerger(theta\_nlin=theta\_nlin, theta\_lin=theta\_lin, x=GQmerger, va=va, n=n, brandFE=T)

c(mean(mer[[1]]), quantile(mer[[1]], 0.5), sd(mer[[1]]))

c(mean(mer[[2]]), quantile(mer[[2]], 0.5), sd(mer[[2]]))

###########################################################

###########################################################

# Full model (5-7)

theta\_nlin <- c(0, 2, 0, 0, 0.3, 2.2, 0.01, 0.2, 5, 13, -0.2, 1.3, 0, -1, 0, 0, 0.2, 0, 0.3, -0.8)

vars <- seq(1, 20, 1)

vars <- paste("z", vars, sep="")

Z <- data.matrix(cbind(x[,vars], dummy.data.frame(as.data.frame(x$brand))))

params <- optim(par=theta\_nlin, fn=blp, x=x, n=n, brand=T, iv=T, supply=F, A=(t(Z)%\*%Z),

method="Nelder-Mead", control=list(reltol=0.1, trace=T))

theta\_nlin <- params$par

theta\_nlin

blp(theta\_nlin=theta\_nlin, x=x, n=n, brand=T, iv=T, supply=F, A=(t(Z)%\*%Z))

coefs

se

c(mean(b), quantile(b, 0.5), sd(b))

c(mean(mc), quantile(mc, 0.5), sd(mc))

nam <- names(coefficients(mean\_u)[2:length(coefficients(mean\_u))])

nam <- gsub("brand", "", nam)

resmd <- data.frame(nam, resmd)

colnames(resmd) <- c("brand", "resmd")

#Post-Nabisco merger

PNmerger <- x

PNmerger$xi <- residuals(mean\_u)

PNmerger$mc <- mc

PNmerger[which(PNmerger$firm==3), "firm"] <- 6

PNmerger$firm <- as.factor(as.character(PNmerger$firm))

PNmerger <- join(PNmerger, resmd, by="brand")

mer <- blpmerger(theta\_nlin=theta\_nlin, theta\_lin=theta\_lin, x=PNmerger, va=va, n=n, brandFE=T)

c(mean(mer[[1]]), quantile(mer[[1]], 0.5), sd(mer[[1]]))

c(mean(mer[[2]], na.rm=T), quantile(mer[[2]], 0.5, na.rm=T), sd(mer[[2]], na.rm=T))

#GM-Quaker merger

GQmerger <- x

GQmerger$xi <- residuals(mean\_u)

GQmerger$mc <- mc

GQmerger[which(GQmerger$firm==2), "firm"] <- 4

GQmerger$firm <- as.factor(as.character(GQmerger$firm))

GQmerger <- join(GQmerger, resmd, by="brand")

mer <- blpmerger(theta\_nlin=theta\_nlin, theta\_lin=theta\_lin, x=GQmerger, va=va, n=n, brandFE=T)

c(mean(mer[[1]]), quantile(mer[[1]], 0.5), sd(mer[[1]]))

c(mean(mer[[2]], na.rm=T), quantile(mer[[2]], 0.5, na.rm=T), sd(mer[[2]], na.rm=T))

blp <- function(theta\_nlin, x, brandFE=F, iv=T, supply=T, n, A=NULL){

# arrange the non linear coefficients

sig <- diag(theta\_nlin[1:4])

pi <- cbind(theta\_nlin[5:8], theta\_nlin[9:12], theta\_nlin[13:16], theta\_nlin[17:20])

# place demographic variables in shorter arrays

dj1 <- seq(1, 20, 1)

dj1 <- paste("v", dj1, sep="")

dj2 <- seq(21, 40, 1)

dj2 <- paste("v", dj2, sep="")

dj3 <- seq(41, 60, 1)

dj3 <- paste("v", dj3, sep="")

dj4 <- seq(61, 80, 1)

dj4 <- paste("v", dj4, sep="")

xt <- split(x, as.factor(x$t))

# fixed point algorithm to compute mean utilities

deltas <- NULL

va <- NULL

sij\_a <- NULL

for(i in 1:length(xt)){

xtt <- xt[[i]]

xi <- data.matrix(xtt[,c("constant", "price", "sugar", "mushy")])

d <- data.matrix(xtt[,c(dj1, dj2, dj3, dj4)])

sjo <- c(xtt$share, 1-sum(xtt$share))

va <- rbind(va,t(rnorm(n)))

v <- matrix(rep(t(t(rnorm(n))), ncol(xi)), ncol=n, byrow = TRUE)

del <- matrix(1, nrow=nrow(xi), ncol=1)

sij <- matrix(0, ncol=n, nrow=nrow(xi)+1)

mu <- matrix(0, ncol=n, nrow=nrow(xi))

for(j in 1:nrow(d)){

dj <- rbind(d[j,dj1], d[j,dj2], d[j,dj3], d[j,dj4])

muj <- xi%\*%sig%\*%v + xi%\*%pi%\*%dj

mu <- mu+(muj/(nrow(d)))

u <- del%\*%rep(1, n) + muj

exp\_u <- exp(rbind(u, rep(0, ncol(u))))

sijt <- sweep(exp\_u, 2, colSums(exp\_u),`/`)

sij <- sij + sijt

}

sj <- rowMeans(sij)

delp <- log(sjo) - log(sj) + c(del,1)

dels <- rbind(c(c(del,1)), c(delp))

tol <- dist(dels)

tol <- as.numeric(tol)

repeat{

del <- delp

u <- t(t(del[1:(length(del)-1)]))%\*%rep(1, n) + mu

exp\_u <- exp(rbind(u, rep(0, ncol(u))))

sij <- sweep(exp\_u, 2, colSums(exp\_u),`/`)

sj <- rowMeans(sij)

delp <- log(sjo) - log(sj) + del

dels <- rbind(c(del), c(delp))

tol <- dist(dels)

tol <- as.numeric(tol)

if (tol<1e-14) break

}

deltas <- rbind(deltas, t(t(delp[1:(length(delp)-1)])))

u <- t(t(delp[1:(length(delp)-1)]))%\*%rep(1, n) + mu

exp\_u <- exp(rbind(u, rep(0, ncol(u))))

sij\_a[[i]] <- sweep(exp\_u, 2, colSums(exp\_u),`/`)

}

va <<- va

# mean utility regression

if(iv==T){

vars <- seq(1, 20, 1)

vars <- paste("z", vars, sep="")

vars <- as.name(paste(vars, collapse="+"))

form <- paste("price", vars, sep="~")

p\_h <- predict(lm(as.formula(form), data=x))

if(brandFE==T) { # iv with brand FE

vars <- c("-1", "p\_h", "brand")

} else { # iv without brand FE

vars <- c("p\_h", "sugar", "mushy")

}

vars <- as.name(paste(vars, collapse="+"))

form <- paste("deltas", vars, sep="~")

} else {

if(brandFE==T) { # ols with brand FE

vars <- c("-1", "price", "brand")

} else { # ols without brand FE

vars <- c("price", "sugar", "mushy")

}

vars <- as.name(paste(vars, collapse="+"))

form <- paste("deltas", vars, sep="~")

}

mean\_u <<- lm(as.formula(form), data=x)

nam <- names(coefficients(mean\_u))

nam <- which(nam=="price"|nam=="p\_h")

#minimum distance estimates for brand dummy FE

if(brandFE==T){

ymd <- coefficients(mean\_u)[2:length(coefficients(mean\_u))]

hvcov <- vcov(mean\_u)[2:length(coefficients(mean\_u)),2:length(coefficients(mean\_u))]

ymd <- matrix(c(as.numeric(na.omit(ymd))),nrow=nrow(hvcov), ncol=1)

xmd <- xt[[1]]

xmd <- data.matrix(xmd[,c("constant", "sugar", "mushy")])

hdmd <- solve(t(xmd)%\*%solve(hvcov)%\*%xmd)%\*%t(xmd)%\*%solve(hvcov)%\*%matrix(c(ymd),nrow=nrow(hvcov), ncol=1)

resmd <<- ymd-xmd%\*%hdmd

semd <- sqrt(diag(solve(t(xmd)%\*%solve(hvcov)%\*%xmd)))

coefs <<- c(hdmd[1], coefficients(mean\_u)[nam], hdmd[2:3])

se <<- c(semd[1], sqrt(vcov(mean\_u)[nam,nam]), semd[2:3])

}

# markup and marginal cost estimates

elasticities <- NULL

mc <- NULL

b <- NULL

for(i in 1:length(xt)){

xtt <- xt[[i]]

d <- data.matrix(xtt[,c(dj1, dj2, dj3, dj4)])

aij <- matrix(0, ncol=n, nrow=nrow(xtt))

for(j in 1:nrow(d)){

dj <- rbind(d[j,dj1], d[j,dj2], d[j,dj3], d[j,dj4])

a <- coefficients(mean\_u)[nam] + t(t(xtt$price))%\*%sig[2,2]%\*%va[i,] + t(t(xtt$price))%\*%pi[2,]%\*%dj

aij <- aij+a

}

aij <- aij/nrow(d)

sijt <- sij\_a[[i]]

sijt <- sijt[1:(nrow(sijt)-1),]

sj <- rowMeans(sijt)

ep <- rowMeans(aij\*sijt\*(1-sijt)) #own-price derivatives of demand

ec <- ((-aij)\*sijt)%\*%t(sijt)/n #cross-price derivatives of demand

diag(ec) <- ep

elasticities[[i]] <- ec\*(xtt$price/xtt$share)

su <- summary(xtt$firm)

frm <- NULL

for(k in 1:length(su)) {

frm[[k]] <- matrix(1, nrow=su[k], ncol=su[k])

}

om <- (-1)\*data.matrix(bdiag(frm))\*ec # omega matrix

b <- rbind(b, solve(om)%\*%sj)

mc <- rbind(mc, xtt$price-solve(om)%\*%sj)

}

b <<- b

mc <<- mc

# estimation of pricing equation

if(supply==T) {

if(brandFE==T) {

w <<- lm(mc ~ sugar + mushy + brand + t, data=x)

} else {

w <<- lm(mc ~ sugar + mushy + t, data=x)

}

g <- matrix(c(residuals(mean\_u),residuals(w)), nrow=length(c(residuals(mean\_u),residuals(w))), ncol=1)

ge <<- g

if(iv==T) {

vars <- seq(1, 20, 1)

vars <- paste("z", vars, sep="")

if(brandFE==T) {

Z <- data.matrix(rbind(cbind(x[,vars], dummy.data.frame(as.data.frame(x$brand))),cbind(x[,vars],dummy.data.frame(as.data.frame(x$brand)))))

} else {

Z <- data.matrix(rbind(x[,vars],x[,vars]))

}

} else {

Z <- data.matrix(rbind(dummy.data.frame(as.data.frame(x$brand)),dummy.data.frame(as.data.frame(x$brand))))

}

} else {

g <- residuals(mean\_u)

ge <<- residuals(mean\_u)

if(iv==T) {

vars <- seq(1, 20, 1)

vars <- paste("z", vars, sep="")

if(brandFE==T) {

Z <- data.matrix(cbind(x[,vars], dummy.data.frame(as.data.frame(x$brand))))

} else {

Z <- data.matrix(x[,vars])

}

} else {

if(brandFE==T) {

Z <- data.matrix(dummy.data.frame(as.data.frame(x$brand)))

}

}

}

# criterion function

if(brandFE == T | iv == T) {

gmm <- ((t(g)%\*%Z)%\*%solve(A)%\*%(t(Z)%\*%g))/nrow(xt[[1]])

}

if(brandFE == F & iv == F) {

gmm <- (t(g)%\*%(g))/nrow(xt[[1]])

}

if(length(gmm)==0) {

gmm <- 1e8

}

return(gmm)

}

blpmerger <- function(theta\_nlin, theta\_lin, x, va, n, brandFE=F){

source("equilibrium.R")

brandFE <<- brandFE

# arrange the non linear coefficients

sig <<- diag(theta\_nlin[1:4])

hpi <<- cbind(theta\_nlin[5:8], theta\_nlin[9:12], theta\_nlin[13:16], theta\_nlin[17:20])

dj1 <- seq(1, 20, 1)

dj1 <<- paste("v", dj1, sep="")

dj2 <- seq(21, 40, 1)

dj2 <<- paste("v", dj2, sep="")

dj3 <- seq(41, 60, 1)

dj3 <<- paste("v", dj3, sep="")

dj4 <- seq(61, 80, 1)

dj4 <<- paste("v", dj4, sep="")

xt <- split(x, as.factor(x$t))

# equilibrium after merger

p\_a <- NULL

sj\_a <- NULL

for(i in 1:length(xt)){

xtt <- xt[[i]]

k <<- i

xtt <<- xtt[order(xtt$firm, xtt$id),]

mer <- FixedPoint(Inputs=xtt$price, Function=equilibrium)

p\_a <- rbind(p\_a, t(t(mer$FixedPoint)))

xtt$price <- t(t(mer$FixedPoint))

xi <- data.matrix(xtt[,c("constant", "price", "sugar", "mushy")])

d <- data.matrix(xtt[,c(dj1, dj2, dj3, dj4)])

# predicted market shares

if(brandFE==F){

del <- xi%\*%theta\_lin + xtt$xi

} else {

hbr <- xi[,c(1,3,4)]%\*%theta\_lin[c(1,3,4)] + xtt$resmd

del <- data.matrix(cbind(xi[,2], dummy.data.frame(as.data.frame(xtt$brand))))%\*%c(theta\_lin[2], hbr) + xtt$xi

}

sij <- matrix(0, ncol=n, nrow=nrow(xi)+1)

for(j in 1:nrow(d)){

dj <- rbind(d[j,dj1], d[j,dj2], d[j,dj3], d[j,dj4])

muj <- xi%\*%sig%\*%matrix(rep(va[i,], ncol(xi)), ncol=n, byrow = TRUE) + xi%\*%hpi%\*%dj

u <- del%\*%rep(1, n) + muj

exp\_u <- exp(rbind(u, rep(0, ncol(u))))

sijt <- sweep(exp\_u, 2, colSums(exp\_u),`/`)/nrow(d)

sij <- sij + sijt

}

sj <- rowMeans(sij)

sj\_a <- rbind(sj\_a, sj[1:(length(sj)-1)])

}

return(list(p\_a, sj\_a))

}

equilibrium <- function(p){

xtt$price <- p

xi <- data.matrix(xtt[,c("constant", "price", "sugar", "mushy")])

d <- data.matrix(xtt[,c(dj1, dj2, dj3, dj4)])

# predicted market shares

if(brandFE==F){

del <- xi%\*%theta\_lin + xtt$xi

} else {

hbr <- xi[,c(1,3,4)]%\*%theta\_lin[c(1,3,4)] + xtt$resmd

del <- data.matrix(cbind(xi[,2], dummy.data.frame(as.data.frame(xtt$brand))))%\*%c(theta\_lin[2], hbr) + xtt$xi

}

sij <- matrix(0, ncol=n, nrow=nrow(xi)+1)

for(j in 1:nrow(d)){

dj <- rbind(d[j,dj1], d[j,dj2], d[j,dj3], d[j,dj4])

muj <- xi%\*%sig%\*%matrix(rep(va[k,], ncol(xi)), ncol=n, byrow = TRUE) + xi%\*%hpi%\*%dj

u <- del%\*%rep(1, n) + muj

exp\_u <- exp(rbind(u, rep(0, ncol(u))))

sijt <- sweep(exp\_u, 2, colSums(exp\_u),`/`)/nrow(d)

sij <- sij + sijt

}

sj <- rowMeans(sij)

# predicted markups

aij <- matrix(0, ncol=n, nrow=nrow(xtt))

for(j in 1:nrow(d)) {

dj <- rbind(d[j,dj1], d[j,dj2], d[j,dj3], d[j,dj4])

aij <- aij + theta\_lin[2] + t(t(xtt$price))%\*%sig[2,2]%\*%va[k,] + t(t(xtt$price))%\*%hpi[2,]%\*%dj

}

aij <- aij/nrow(d)

sijt <- sij[1:(nrow(sij)-1),]

ep <- rowMeans(aij\*sijt\*(1-sijt))

ec <- (-aij\*sijt)%\*%t(sijt)/n

diag(ec) <- ep

su <- summary(xtt$firm)

frm <- NULL

for(k in 1:length(su)) {

frm[[k]] <- matrix(1, nrow=su[k], ncol=su[k])

}

om <- (-1)\*(data.matrix(bdiag(frm)))\*ec

bt <- solve(om)%\*%sj[1:(length(sj)-1)]

hp <- xtt$mc+bt

return(hp)

}