ECO7707 - Problem Set IV

Armand Kapllani

4/15/2018

```
#---------------#
# Problem Set 4 #
#---------------#

#-------------------------------------------------------------- #
# For simplicity all data are uploaded on my github account. #
# We will import them directly from there.                   #
#--------------------------------------------------------------#

# Import PS3 dataset.
PS3_data <- read.table(file = "https://raw.githubusercontent.com/armandkapllani/IER-PS4-DATA/master/fina
PS3_data <- data.table(PS3_data)

# Import population data.
population <- read.table(file = "https://raw.githubusercontent.com/armandkapllani/IER-PS4-DATA/master/po
population <- data.table(population)

# Import gdp per capita data.
gdp_capita <- read.table("https://raw.githubusercontent.com/armandkapllani/IER-PS4-DATA/master/gdp_capit
gdp_capita <- data.table(gdp_capita)

# Import gdp from Problem Set I (for year 2014)
gdp <- read.csv("https://raw.githubusercontent.com/armandkapllani/IER-PS4-DATA/master/gdp_2014.csv", sep
gdp <- data.table(gdp)
```

```
#----------------------------------------------------------------------#
# 1. Use your data from problem set 3 but keep only data for 3 goods #
#    1. 0302120003                                                  #
#    2. 2501000000                                                  #
#    3. 8711200090                                                  #
#----------------------------------------------------------------------#

dta_o <- PS3_data[commodity == 0302120003 | commodity == 2501000000 | commodity == 8711200090]
dta <- dta_o

# Check if the job was done correctly.
unique(dta$commodity)
```

```
## [1]  302120003 2501000000 8711200090
```

```
# Estimate this regression using shipping costs as instrument as before and use data for periods 2010- 
# IV Regression: [Instrument: Shipping Costs].

# Construct the instrument
IV <- log(dta$cif_charges/dta$quantity)

# Lets check again for NaN or -Inf, Inf in the IV
IV[which(!is.finite(IV))] = NA
```

```r
dta <- data.table(dta, IV) %>% na.omit()

# Create import shares.
dta <- dta[quantity != 0]
dta <- dta[, TotalSum := sum(quantity), by = .(commodity, year)]
dta <- dta[, share := quantity/TotalSum]

# IV regression for each commodity.

# Commodity: 0302120003
reg1 <- ivreg(log(share) ~ 0 + factor(cty_code) + log(price) | . -log(price) + IV, data = dta[commodity

rank_reg1 <- cbind(setnames(data.table(head(substr(names(coef(reg1)), 17, 20), -1)),'V1', 'cty_code'),
                   setnames(data.table(head(coef(summary(reg1))[,1], -1)), 'V1', 'est'))

# Sort them
rank_reg1[order(-rank(est), cty_code)][1:3]
```

```
##    cty_code      est
## 1:     1220 7.407389
## 2:     4091 6.876552
## 3:     4120 6.835720
```

```r
rank_reg1[order(rank(est), cty_code)][1:3]
```

```
##    cty_code       est
## 1:     4279 -2.695457
## 2:     4210 -1.087562
## 3:     3310 -1.075304
```

```r
# Three highest quality producers:
# 1. Canada
# 2. Faroe Islands
# 3. United Kingdom

# Three lowest quality producers
# 1. France
# 2. Netherlands
# 3. Ecuador

# Commodity: 2501000000
reg2 <- ivreg(log(share) ~ 0 + factor(cty_code) + log(price) | . -log(price) + IV, data = dta[commodity

rank_reg2 <- cbind(setnames(data.table(head(substr(names(coef(reg2)), 17, 20), -1)),'V1', 'cty_code'),
                   setnames(data.table(head(coef(summary(reg2))[,1], -1)), 'V1', 'est'))

# Sort them
rank_reg2[order(-rank(est), cty_code)][1:3]
```

```
##    cty_code      est
## 1:     3370 2.840277
## 2:     1220 2.583296
## 3:     2010 1.397456
```

```r
rank_reg2[order(rank(est), cty_code)][1:3]
```

```
##    cty_code       est
## 1:     4050 -8.359076
## 2:     2720 -8.151200
## 3:     7880 -8.059257
```

```
# Three highest quality producers:
#1. Chile
#2. Canada
#3. Mexico

# Three lowest quality producers
#1. Finland
#2. Barbados
#3. Madagascar

# Commodity: 8711200090
reg3 <- ivreg(log(share) ~ 0 + factor(cty_code) + log(price) | . -log(price) + IV, data = dta[commodity

rank_reg3 <- cbind(setnames(data.table(head(substr(names(coef(reg3)), 17, 20), -1)),'V1', 'cty_code'),
                   setnames(data.table(head(coef(summary(reg3))[,1], -1)), 'V1', 'est'))

# Sort them
rank_reg3[order(-rank(est), cty_code)][1:3]
```

```
##    cty_code      est
## 1:     5880 7.457507
## 2:     5490 6.248741
## 3:     4330 6.070887
```

```
rank_reg3[order(rank(est), cty_code)][1:3]
```

```
##    cty_code       est
## 1:     4190 -2.723156
## 2:     4351 -2.208382
## 3:     4550 -2.020141
```

```
# Three highest quality producers:
#1. Japan
#2. Thailand
#3. Austria

# Three lowest quality producers
#1. Ireland
#2. Czech Republic
#3. Poland
```

```
#-----------------------------------------------------------------------------------------------#
# 2. Add information on population to the dataset by using merge() and the file population.csv  #
#    Remove observations for which there is no population data given.                           #
#-----------------------------------------------------------------------------------------------#

dta <- merge(dta, population, by = c("cty_code","year"))
setnames(dta, "population", "pop")
```

```
#-------------------------------------------------------------------------------------------------#
# 3. Add log(population) as control in the above regressions and state the 3 highest and 3 lowest #
#    quality producers in each category now. Do they appear more or less reasonable to you?       #
```

```r
#----------------------------------------------------------------------------------------------------#

# IV regression for each commodity [controlling for population]

# Commodity: 0302120003 [controlling for population]
reg11 <- ivreg(log(share) ~ 0 + factor(cty_code) + log(pop) + log(price) | . -log(price) + IV, data = df

rank_reg11 <- cbind(setnames(data.table(head(substr(names(coef(reg11)), 17, 20), -2)),'V1', 'cty_code')
                    setnames(data.table(head(coef(summary(reg11))[,1], -2)), 'V1', 'est'))

# Sort them highest to lowest.
rank_reg11[order(-rank(est), cty_code)][1:3]
```

```
##    cty_code      est
## 1:     4120 326.5797
## 2:     4279 317.0510
## 3:     1220 309.6333
```

```r
# Three highest quality producers.
#1. United Kingdom
#2. France
#3. Canada

# Sort them lowest to highest.
rank_reg11[order(rank(est), cty_code)][1:3]
```

```
##    cty_code      est
## 1:     4000 165.9424
## 2:     6141 247.9184
## 3:     4190 248.6683
```

```r
# Three lowest quality producers.
#1. Iceland
#2. New Zealand
#3. Ireland

# Commodity: 2501000000 [controlling for population]
reg21 <- ivreg(log(share) ~ 0 + factor(cty_code) + log(pop) + log(price) | . -log(price) + IV, data = df

rank_reg21 <- cbind(setnames(data.table(head(substr(names(coef(reg21)), 17, 20), -2)),'V1', 'cty_code')
                    setnames(data.table(head(coef(summary(reg21))[,1], -2)), 'V1', 'est'))

# Sort them highest to lowest.
rank_reg21[order(-rank(est), cty_code)][1:3]
```

```
##    cty_code       est
## 1:     6830 -18.17090
## 2:     2779 -18.76568
## 3:     2360 -19.67083
```

```r
# Three highest quality producers.
#1. Palau
#2. Aruba
#3. Bahamas
```

```r
# Sort them from lowest to highest.
rank_reg21[order(rank(est), cty_code)][1:3]
```

```
##    cty_code       est
## 1:     5330 -51.31218
## 2:     5700 -49.88086
## 3:     5600 -49.10862
```

```r
# Three lowest quality producers.
#1. India
#2. China
#3. Indonesia

# Commodity: 8711200090 [controlling for population]
reg31 <- ivreg(log(share) ~ 0 + factor(cty_code) + log(pop) + log(price) | . -log(price) + IV, data = d


rank_reg31 <- cbind(setnames(data.table(head(substr(names(coef(reg31)), 17, 20), -2)),'V1', 'cty_code')
                    setnames(data.table(head(coef(summary(reg31))[,1], -2)), 'V1', 'est'))

# Sort them from highest to lowest.
rank_reg31[order(-rank(est), cty_code)][1:3]
```

```
##    cty_code       est
## 1:     4330 -70.60963
## 2:     6141 -72.56929
## 3:     4190 -74.30605
```

```r
# Three highest quality producers.
#1. Austria
#2. New Zealand
#3. Ireland

# Sort them from lowest to highest.
rank_reg31[order(rank(est), cty_code)][1:3]
```

```
##    cty_code       est
## 1:     5330 -121.0809
## 2:     5700 -114.8923
## 3:     5600 -104.8914
```

```r
# Three lowest quality producers.
#1. India
#2. China
#3. Indonesia


# Yes they appear to be more or less reasonable. The results are consistent with the
# Khandelwal (2010) model's prediction that more advanced countries will manufacture
# higher quality products.
```

```r
#------------------------------------------------------------------------------------------#
# 4. Regress the quality estimates you get in (3) on the respective prices of the varieties. #
#    Are higher-quality varieties more expensive?                                          #
#------------------------------------------------------------------------------------------#
```

```
# For regression reg11

# Retrieve cty_code
coef_name1 <- data.table(substr(names(coef(reg11)), 17, 20))
coef_name1 <- head(coef_name1,-2)
setnames(coef_name1, 'V1', 'cty_code')

# Retrieve the estimates for each cty_code
coef_est1 <- head(data.table(coef(reg11)), -2)
setnames(coef_est1, 'V1', 'est1')

# cbind cty_code and respective estimates
dta1 <- cbind(coef_name1, coef_est1)
dta1$cty_code <- as.numeric(as.character(dta1$cty_code))

# For regression reg21

# Retrieve cty_cpde
coef_name2 <- data.table(substr(names(coef(reg21)), 17, 20))
coef_name2 <- head(coef_name2,-2)
setnames(coef_name2, 'V1', 'cty_code')

# Retrieve the estimates for each cty_code
coef_est2 <- head(data.table(coef(reg21)), -2)
setnames(coef_est2, 'V1', 'est2')

# cbind cty_code and respective estimates
dta2 <- cbind(coef_name2, coef_est2)
dta2$cty_code <- as.numeric(as.character(dta2$cty_code))

# For regression reg31
coef_name3 <- data.table(substr(names(coef(reg31)), 17, 20))
coef_name3 <- head(coef_name3,-2)
setnames(coef_name3, 'V1', 'cty_code')

# Retrieve the estimates for each cty_code
coef_est3 <- head(data.table(coef(reg31)), -2)
setnames(coef_est3, 'V1', 'est3')

# cbind cty_code and respective estimates
dta3 <- cbind(coef_name3, coef_est3)
dta3$cty_code <- as.numeric(as.character(dta3$cty_code))

# Now merge each one of them to a new dataset which includes only each of three varieties

# Regress estimates of quality on prices for good1
good1 <- dta[commodity == 0302120003]
dta1 <- merge(dta1, good1, by = 'cty_code')

reg_good1 <- lm(est1 ~ log(price), data = dta1)

# Regress estimates of quality on prices for good2
good2 <- dta[commodity == 2501000000]
```

```r
dta2 <- merge(dta2, good2, by = 'cty_code')

# Regress estimates of quality on prices for good2
reg_good2 <- lm(est2 ~ log(price), data = dta2)


good3 <- dta[commodity == 8711200090]
dta3 <- merge(dta3, good3, by = 'cty_code')

# Regress estimates of quality on prices
reg_good3 <- lm(est3 ~ log(price), data = dta3)



# We see that the estimated coefficient for the motorcycle commodity is the only one that has a
# positive value which shows that higher quality varieties are more expensive. While for commo-
# dity salt and atlantic salmon the estimated coefficient on log(price) is negative, and
# especially the estimated coefficient for salt is very high in absolute value. However they are
# statistically insignificant.
```

```r
#----------------------------------------------------------------------------------------------#
# 5. Regress the quality estimates you get in (3) on the respective countries' income per capita #
# (in logs) using the file gdp capita.csv for each product category separately. Do richer       #
# countries producer higher-quality varieties?                                                 #
#----------------------------------------------------------------------------------------------#

dta_gdp_pc1 <- merge(dta1, gdp_capita, by = c("cty_code","year"))
dta_gdp_pc2 <- merge(dta2, gdp_capita, by = c("cty_code","year"))
dta_gdp_pc3 <- merge(dta3, gdp_capita, by = c("cty_code","year"))

reg_gdp_pc1 <- lm(est1 ~ log(gdp_per_capita), data = dta_gdp_pc1)


reg_gdp_pc2 <- lm(est2 ~ log(gdp_per_capita), data = dta_gdp_pc2)


reg_gdp_pc3 <- lm(est3 ~ log(gdp_per_capita), data = dta_gdp_pc3)



# Yes as we can see from the regression results richer countries produce higher-quality varieties.
# The estimated coefficient on gdp per capita for commodity motorcycles is positive (9.069) showing
# that as GDP per capita increases that will lead to an increase in the production of high quality
# products. Also the estimated coefficient on gdp per capita for coomodity salt is also
# positive. While the estimated coefficient on gdp per capita for the atlantic salmon is statistically
# insignificant.
```

```r
#----------------------------------------------------------------------------------------------------
# 6. Download data from county business patterns (https://www.census.gov/programs- surveys/cbp.html) on
#    U.S. employment for salt and motorcycles. Compute the percentage change in employment in these
#    industries between 1998 and 2014.
#----------------------------------------------------------------------------------------------------

# Note: After 2007, the Census reports data for employment on each sector (naics) and each legal form o
```

```r
#       In the data they are defined as follows:

# '-' - All Establishments
# C - Corporations
# Z - S-Corporations
# S - Sole Proprietorships
# P - Partnerships
# N - Non-Profits
# G - Government
# O - Other

# For our analysis we use the data on employment for 'All Establishments'.
# Technical document:
# https://www2.census.gov/programs-surveys/rhfs/cbp/technical%20documentation/2015_record_layouts/us_la

# Download the zip file from the website directly as follows:
temp <- tempfile()
download.file("https://www2.census.gov/programs-surveys/cbp/datasets/2014/cbp14us.zip",temp)
dta2014 <- read.table(unz(temp, "cbp14us.txt"), header = TRUE, sep = ",")
unlink(temp)  # Remove the temp file

# Download the .txt file from the website.
dta1998 <- read.table("https://www2.census.gov/programs-surveys/cbp/datasets/1998/cbp98us.txt", header=

dta2014<- data.table(dta2014)
dta2014 <- dta2014[(naics == 212393 | naics == 336991) & lfo == "-"]

dta1998 <- data.table(dta1998)
dta1998 <- dta1998[naics == 212393 | naics == 336991]

# Compute the percentage change in employment in these industries between 1998 and 2014.
delta_98_14_salt <- (dta2014$emp[1] - dta1998$emp[1])/dta1998$emp[1]*100
delta_98_14_salt
```

```
## [1] -5.18111
```

```r
delta_98_14_motor <- (dta2014$emp[2] - dta1998$emp[2])/dta1998$emp[2]*100
delta_98_14_motor
```

```
## [1] -30.26021
```

```r
#-----------------------------------------------------------------------------------------------#
# 7. Khandelwal (2010) finds that import competition from low-wage countries has had a negative #
#    impact on U.S. employment but less so in industries with longer quality ladders. Based on  #
#    your previous estimates, compute the quality ladder for the 2 goods as well as the import  #
#    penetration ratio for China. Would you say the results for the 2 categories here match the #
#    paper's findings?                                                                          #
#-----------------------------------------------------------------------------------------------#
# Compute the quality ladder and import penetration for: 2501000000 (salt)
# From part three we derived the quality ladders:

ql_salt <- max(dta2$est2) - min(dta2$est2)
ql_salt
```

```
## [1] 33.14128
```

```r
# Compute the quality ladder and import penetration for: 8711200090 (motorcycles)

ql_motor <- max(dta3$est3) - min(dta3$est3)
ql_motor
```

```
## [1] 50.47129
```

```r
# Import penetrations.
# Import penetration from China for salt: 2501000000

import_usa_china_salt <- PS3_data[year==2014 & commodity == 2501000000 &  cty_code == 5700][,'gross_valu
import_usa_world_salt <- sum(PS3_data[year==2014 & commodity == 2501000000][,'gross_value'])

IP_China_salt <- (import_usa_china_salt/import_usa_world_salt)*100
setnames(IP_China_salt, 'gross_value', 'IP')
IP_China_salt
```

```
##           IP
## 1: 0.3638371
```

```r
# Import penetration from China for motorcycles: 8711200090

import_usa_china_motor <- PS3_data[year==2014 & commodity == 8711200090 &  cty_code == 5700][,'gross_val
import_usa_world_motor <- sum(PS3_data[year==2014 & commodity == 8711200090][,'gross_value'])

IP_China_motor <- (import_usa_china_motor/import_usa_world_motor)*100
setnames(IP_China_motor, 'gross_value', 'IP')
IP_China_motor
```

```
##         IP
## 1: 4.74604
```

```r
# Another method: import_salt_from_china/ (import_salt_from_china + production_salt_usa - usa_export_to
# For this method we use Comtrade data and BEA GDP by Industry.

# USA production of salt: $24,212,000       (BEA: GDP by Industry, using NAICS)
# USA export of salt to China: $15,804,069  (BEA dataset)

IP_China_salt_a <- (import_usa_china_salt/(import_usa_china_salt + 24212000 - 15804069))*100
setnames(IP_China_salt_a, 'gross_value', 'IP')
IP_China_salt_a
```

```
##          IP
## 1: 24.83919
```

```r
# USA production of motorcycles: $6,460,000
# USA export of motorcycles to China: $1,077,916

IP_China_motor_a <- (import_usa_china_motor/(import_usa_china_motor + 6460000 - 1077916))*100
setnames(IP_China_motor_a, 'gross_value', 'IP')
IP_China_motor_a
```

```
##          IP
## 1: 60.12502
```

```r
#-------------#
# ENTRY GAMES #
```

```r
#-------------#

#--------------------------------------------------------------------------#
# Jia (2008) algorithmic approach in determining the supremum and infimum #
#--------------------------------------------------------------------------#
```

```r
#---------------------------------------------------------------------------------------------------#
# 1. The file data estimation.csv provides information on each county's population and a dummy that   #
#    is 1 if it is in the south. The file dist.RData provides the distances between each city pair.1. #
#    Using Jia's (2008) algorithm, find the least element in the set of fixed points DL.               #
#---------------------------------------------------------------------------------------------------#

# Import estimation.csv from git.
estimation <- read.table("https://raw.githubusercontent.com/armandkapllani/IER-PS4-DATA/master/Data_Est

# Import dist.RData from git.
download.file("https://github.com/armandkapllani/IER-PS4-DATA/blob/master/dist.RData?raw=true", "dist")
load("dist")


# Right each element in the dist matrix as 1/z
# Make diagonal element in the matrix equal to zero.
Z <- dist
Z_i <- 1/Z
diag(Z_i) <- 0

# Let P be the log population vector. (380x1)
P <- matrix(log(estimation$population), ncol = 1)
estimation$P <- P

# Let S be the south vector. (380x1)
S <- estimation$south

# Create a vector 380x1 of ones.
one <- matrix(rep(1, nrow(Z_i)), ncol = 1)

# Vector of marginal benefits
Pi <- matrix(0, nrow = 380)

#---------------------------------------------------#
# Find the supremum.                                 #
#---------------------------------------------------#

D_init<- matrix(rep(1, nrow(Z_i)), ncol = 1)  # set initial vector of ones.
D_new <- D_init
iter <- 1

repeat {

  D_old = D_new
  Pi = -55*one + 5*P - 2*S + 0.1*Z_i%*%D_old
  D_new = ifelse(Pi > 0, 1, 0)

  if (isTRUE(all.equal(D_new, D_old))){
```

```r
    break
  }
  iter = iter + 1
}

DU <- D_new     # supremum vector.

#--------------------------------------------------#
# Find the infimum.                                #
#--------------------------------------------------#

D_init<- matrix(rep(0, nrow(Z_i)), ncol = 1)  # set initial vector of zeros.
D_new <- D_init
iter <- 1

repeat {

  D_old = D_new
  Pi = -55*one + 5*P - 2*S + 0.1*Z_i%*%D_old
  D_new = ifelse(Pi > 0, 1, 0)

  if (isTRUE(all.equal(D_new, D_old))){
    break
  }
  iter = iter + 1
}

DL <- D_new   # infimum vector.
```

```r
#----------------------------------------#
# 3. Which elements differ in DL and DU? #
#----------------------------------------#

# Check that DU greater than DL (just making sure that Tarski(1955) was right)
summary(DU>=DL)
```

```
##      V1
##  Mode:logical
##  TRUE:380
```

```r
# Append both supremum vector and infimum vector to estimation data.
estimation <- cbind(estimation, DL, DU)
estimation <- data.table(estimation)

# Denote by "Yes" if an element i of vector DL differs from an element i of vector DU.
for(i in 1:nrow(estimation)){
  if(DL[i] != DU[i]){
    estimation$diff[i] <- 'Yes'
  }
  else
    estimation$diff[i] <- 'No'
}

estimation[, c('DL', 'DU', 'diff')]
```

```
##      DL DU diff
##   1:  0  1  Yes
##   2:  0  1  Yes
##   3:  0  0   No
##   4:  1  1   No
##   5:  1  1   No
##  ---
## 376:  1  1   No
## 377:  1  1   No
## 378:  1  1   No
## 379:  1  1   No
## 380:  1  1   No
```

```
# Count the number of elements that differ and show the counties that differ only.
estimation[, .N, by = diff]
```

```
##    diff   N
## 1:  Yes  14
## 2:   No 366
```

```
estimation[diff == 'Yes'][,'County']
```

```
##               County
##  1:     Arkansas, AR
##  2:       Ashley, AR
##  3:     Franklin, AR
##  4: Little River, AR
##  5:         Pope, AR
##  6:     San Juan, CO
##  7:     Crawford, IL
##  8:   Cumberland, IL
##  9:    Henderson, IL
## 10:   Jo Daviess, IL
## 11:     Moultrie, IL
## 12:      Buffalo, WI
## 13:      Burnett, WI
## 14:       Forest, WI
```

```
# So now the problem we have to solve is much easier.
# Hence we now consider only 2^14 = 16384 possible combinations or 10^4.21441995 combinations.
```

```
#----------------------------------------------------------------------------------------------#
# 4. Find Wal-Mart's actual decision, i.e. which counties it optimally enters. Remember that #
#    the solution will be between DL and DU so you only need to evaluate which solution in   #
#    this subset maximizes Wal-Mart's profit.                                                 #
#----------------------------------------------------------------------------------------------#

# All possible combinations for the ones who are different 2^14 = 16384.
m <- data.frame(t(do.call(CJ, replicate(14, 0:1, FALSE))))

# Specify the indicies for the which the values differ.
indeces <- which(estimation$diff == 'Yes')

# Set row names of the combinations matrix.
rownames(m) <- indeces
```

```r
# We evaluate each possible combination above and compute the profits for each combination
comb <- matrix(estimation$DL, nrow = 380, ncol = 2^14)
rownames(comb) <- 1:380

# Insert all possible combinations created in m matrix in the comb matrix indeces where
# the values 14 values were different. [THIS WILL TAKE 67.498 sec]
for(j in 1:ncol(comb)){
  for(i in rownames(m)){
    comb[i,j] = m[i,j]
  }
}

# Compute the profits for each possible combination.
Pi_b <- matrix(0, nrow = nrow(comb), ncol = ncol(comb))
for(c in 1:ncol(comb)){
  Pi_b[,c] = -55*one + 5*P - 2*S + 0.05*Z_i%*%comb[,c]
}

# Now sum over the columns for each combination find the maximum profit and
# its respective index.
profits <- matrix(colSums(Pi_b), ncol = 1)

# Maximum profit
max(profits)
```

```
## [1] 1547.626
```

```r
# Index of the vector that maximizes the profits.
which.max(profits)
```

```
## [1] 16384
```

```r
# Which vector?
max_vector <- matrix(comb[,which.max(profits)], nrow(comb))

# Which counties should we enter?
estimation <- data.table(estimation, max_vector)
setnames(estimation, "V1", "max_vector")
estimation[max_vector == 1][,"County"]
```

```
##             County
##   1:  Arkansas, AR
##   2:    Ashley, AR
##   3:    Benton, AR
##   4:     Boone, AR
##   5: Craighead, AR
##  ---
## 219:  Waukesha, WI
## 220:   Waupaca, WI
## 221:  Waushara, WI
## 222: Winnebago, WI
## 223:      Wood, WI
```

```r
# Lets check if this vector is the same as our supremum vector
summary(max_vector == DU)
```

```
##      V1
##   Mode:logical
##   TRUE:380
```

```
# So in this case the supremum that we found before is the combinations vector that maximizes
# our profits.
```

```
#-----------------------------------------------------------------------------------------------#
# 5. The profit function above assumes that Wal-Mart's profits are lower in the south. This will #
#    likely not be the case given Jia's (2008) results. Suppose all other parameters are true,   #
#    describe briefly how you would estimate the true coefficient on South.                      #
#-----------------------------------------------------------------------------------------------#
```

Given the profit function

$$\pi_m = \gamma_0 + \gamma_1 \ln(pop)_m + \gamma_2 South_m + \gamma_3 \sum_{l \neq m} \frac{D_l}{D_m}$$

We assume that $(\gamma_0, \gamma_1, \gamma_3) = (-55, 5, 0.05)$ are the true parameters and we need to estimate only γ_2.

Using the simulated method of moments we can estimate γ_2 as follows:

1. The first thing we need to do is to set up the moment or moments that will identify our parameter of interest γ_2.
2. Start with an initial guess of γ_2 and solve the model with that parameter.
3. Then compute the difference between the actual moment/moments in the data with the simulated one $\hat{m}(\gamma_2)$.

$$\hat{g}(\gamma_2) = m - \hat{m}(\gamma_2)$$

4. Repeat steps 1-3 for a different set of parameters of γ_2 until you will find one such that it will minimize the sum of squares of these differences.

$$\gamma_2 := \arg\min \hat{g}(\gamma_2)' \hat{g}(\gamma_2)$$

```
#----------------------------------------#
# Some extra work for no extra points :) #
#----------------------------------------#
#---------------------------------------------------------------------------------------#
# I Love for-loops, so this was my first solution for finding the infimum and supremum.  #
# Not very elegant though when compared to the previous one but it produces the same     #
# results.                                                                               #
#---------------------------------------------------------------------------------------#

#---------------#
# Finds infimum #
#---------------#

D_init <- matrix(rep(0, nrow(Z_i)), ncol = 1)
D_old <- D_init
iter <- 1

repeat {
  D_new = D_old
  for(i in 1:nrow(D_old)){
```

```r
    Pi[i] = -55*one[i] + 5*P[i] - 2*S[i] + 0.1*Z_i[i,]%*%D_old
    if(Pi[i] > 0){
      D_new[i] = 1
    }
    else if(Pi[i] <= 0){
      D_new[i] = 0
    }
  }
  if (isTRUE(all.equal(D_new, D_old))){
    break
  }
  D_old <- D_new
  iter <- iter + 1
}

infimum = D_old

#----------------#
# Finds supremum #
#----------------#

D_init <- matrix(rep(1, nrow(Z_i)), ncol = 1)

D_old <- D_init
iter <- 1

repeat {
  D_new = D_old
  for(i in 1:nrow(D_old)){
    Pi[i] = -55*one[i] + 5*P[i] - 2*S[i] + 0.1*Z_i[i,]%*%D_old
    if(Pi[i] > 0){
      D_new[i] = 1
    }
    else if(Pi[i] <= 0){
      D_new[i] = 0
    }
  }
  if (isTRUE(all.equal(D_new, D_old))){
    break
  }
  D_old <- D_new
  iter <- iter + 1
}

supremum <- D_old
```