Econ821 Problem Set 2 Result Summaries

Xuan Lin

The raw data was firstly transferred to a long data frame on the python platform and sorting models were implemented in R. Hope this version can provide a possibility to compare how results in R differ from other programming languages in sorting analyses.

Regarding sorting models programming on R:

- The package called "mlogit" can deal with those simple multinominal conditional logit models when the BLP contraction mapping is not required. It is very easy to implement and the function form is the only input we need. Results are quite comparable.
- I wrote the codes for BLP contract mapping. The iteration process is quick and results are comparable.
- I am still trying to figure out the programming of quantile IV estimation described in Timmins and Murdock (2007). It is one of the difficulties I have encountered. To my knowledge, some people on github and R platform were working on this but I haven't found a good solution yet. If there is any other code versions on Matlab and Python I can have a look, I am quite interested in code transferring and trying the model on R. Much appreciated in advanced! (2sls and another quantile estimation were used in 3.ii instead)

Question 2.i

Ignoring difference in individual characteristics and unobserved site attributes

```
7 iterations, 0h:0m:4s
g'(-H)^-1g = 1.08E-05
successive function values within tolerance limits

Coefficients :
             Estimate Std. Error  z-value  Pr(>|z|)
ramp        0.0431794  0.0835485   0.5168  0.605283
restroom   -0.1829116  0.0670463  -2.7281  0.006369 **
walleye     1.7361704  0.1131567  15.3431 < 2.2e-16 ***
salmon      4.6005104  0.2915085  15.7817 < 2.2e-16 ***
panfish     0.3859849  0.0335145  11.5169 < 2.2e-16 ***
travelcost -0.1030538  0.0016821 -61.2657 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log-Likelihood: -5820.5
```

Question 2.ii

Includes interaction terms on preferences

```
7 iterations, 0h:0m:6s
g'(-H)^-1g = 1.19E-05
successive function values within tolerance limits

Coefficients :
                Estimate Std. Error  z-value  Pr(>|z|)
ramp          -0.4118051  0.1050004  -3.9219 8.784e-05 ***
restroom      -0.2778554  0.0785884  -3.5356 0.0004069 ***
walleye        1.2992691  0.1798558   7.2239 5.049e-13 ***
salmon         4.5561978  0.2915801  15.6259 < 2.2e-16 ***
panfish        0.4378603  0.0381338  11.4822 < 2.2e-16 ***
travelcost    -0.1030564  0.0016877 -61.0618 < 2.2e-16 ***
panfish_kids  -0.1632874  0.0557749  -2.9276 0.0034157 **
restroom_kids  0.3442013  0.1339883   2.5689 0.0102025 *
ramp_boat      0.9818902  0.1617496   6.0704 1.276e-09 ***
walleye_boat   0.6961032  0.2152914   3.2333 0.0012237 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Question 2.iii

Includes an unobserved site attributes

First stage: Contraction Mapping

```
> print (maxlikelihood)
[1] -5136.956
> print(a)
   travelcost  panfish_kids restroom_kids     ramp_boat  walleye_boat
   -0.1230331    -0.1408312     0.5068775     1.3974301     0.5928853
```

Second stage: OLS

```
> BLP2 <- lm(theta_j ~ ramp + restroom + walleye + salmon + panfish + 0, data=data)
> summary(BLP2)

Call:
lm(formula = theta_j ~ ramp + restroom + walleye + salmon + panfish +
    0, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-2.8631 -0.9891 -0.1809  0.8354  3.4372

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
ramp     -0.582122   0.007771  -74.91   <2e-16 ***
restroom -0.735605   0.006926 -106.21   <2e-16 ***
walleye   2.008498   0.013643  147.22   <2e-16 ***
salmon    2.221400   0.036231   61.31   <2e-16 ***
panfish   0.498073   0.002991  166.50   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.327 on 240395 degrees of freedom
Multiple R-squared:  0.1684,	Adjusted R-squared:  0.1684
F-statistic:  9735 on 5 and 240395 DF,  p-value: < 2.2e-16
```

Question 2.iv

My codes didn't give a convergent result. Still trying..

Question 2 interpretation

- (a) travel cost to site and restroom reduce utilities. Other factors such as the presence of ramp, expected catch rate of walleye, salmon and panfish contribute to greater utility.
- (b) When we include interaction terms between individual characteristics and site attributes, coefficients of interest do vary. For example, coefficient of ramp turns to be negative, but ramp would benefit people who are with boats. Coefficient of restroom become more negative, but

anglers appreciate the presence of restroom if they bring kids. Coefficients of walleye, salmon and panfish experience small changes, while results suggest that people with kids dislike a higher catch rate of panfish and the preference for walleye catching rate is higher if people bring their boats, suggesting a more enjoyable of walleye catching with boats. The coefficients of travel cost keep nearly same.

- (c) The second model illustrates the importance of accounting for the observable difference among individuals, to some extent, correcting the biases of first model.

- (d) Including unobserved site attributes didn't give a qualitatively different conclusion, since the sign of all coefficients we measure didn't flip. However, there are some noticeable change in the magnitude of coefficient estimations that would be interesting for interpretation. The marginal utility of travel cost further declined from -0.103 to -0.123, suggesting that the upward-biased estimation of first two models are captured by the unobserved site attributes. Some utility brought by unobserved site attributes compensate for the dis-utility of long-distance travelling. Less congestion level could be a possible explanation: People travel far to exchange for less crowded site. Also, there must be some correlation between unobserved attributes and the presence of restroom, because the corresponding coefficient change dramatically from -2.779 to -0.736. e.g. congestion makes bathrooms be more crowded and awful to use.

Question 3.i

repeat 2.i but including the share attribute

```
Coefficients :
             Estimate Std. Error  z-value  Pr(>|z|)
ramp       -0.0077524  0.0841435  -0.0921  0.926592
restroom   -0.2249888  0.0693788  -3.2429  0.001183 **
walleye     0.1732772  0.1348250   1.2852  0.198722
salmon      0.2381518  0.3842951   0.6197  0.535448
panfish     0.2307498  0.0353148   6.5341   6.4e-11 ***
travelcost -0.1044365  0.0017152 -60.8891 < 2.2e-16 ***
shares_100  0.6940338  0.0291272  23.8277 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log-Likelihood: -5511.3
```

Interpretation: When we include the measurement of congestion into the model, the marginal utility of site attributes generally decline and the measurement of coefficients of walleye and salmon even turn out to be insignificant. Th effect of travel cost is similar.

repeat 2.ii but including the share attribute

```
Coefficients :
               Estimate Std. Error  z-value  Pr(>|z|)    
ramp          -0.509431   0.106758  -4.7718 1.826e-06 ***
restroom      -0.320070   0.081220  -3.9408 8.122e-05 ***
walleye       -0.205567   0.195433  -1.0519  0.292867    
salmon         0.184088   0.385202   0.4779  0.632721    
panfish        0.267856   0.039444   6.7908 1.115e-11 ***
travelcost    -0.104440   0.001722 -60.6494 < 2.2e-16 ***
panfish_kids  -0.126635   0.055352  -2.2878  0.022148 *  
restroom_kids  0.374430   0.138577   2.7020  0.006893 ** 
ramp_boat      1.048108   0.162745   6.4402 1.193e-10 ***
walleye_boat   0.594764   0.219561   2.7089  0.006751 ** 
shares_100     0.693968   0.029186  23.7774 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-Likelihood: -5477.2
```

Interpretation: Similar to the result of fist model, when the preference heterogeneity and congestion level are both included in the estimation model, the coefficient measurement of the marginal utility of salmon and walleye become insignificant. The marginal utility of other site attributes become smaller and that of walleye even turn to be negative, suggesting that the consideration of congestion in fishing sites could make other site attributes less important and even don't matter to some extent. Congestion is an influential determinant of anglers' utility and decision making. The interaction terms tend to have lower marginal utility besides $ramp * boat$, but the change in magnitude is small.

repeat 2.iii but including the share attribute

First stage: Contraction Mapping

```
> print(a)
   travelcost  panfish_kids restroom_kids     ramp_boat  walleye_boat 
   -0.1230306    -0.1408963     0.5048161     1.3881009     0.5524893 
> print (maxlikelihood)
[1] -5136.971
```

Second stage: OLS result

```
Call:
lm(formula = theta_j ~ ramp + restroom + walleye + salmon + panfish +
    shares_100 + 0, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-2.7653 -0.9529 -0.0086  0.9183  3.6131

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
ramp       -0.685585   0.007276  -94.23   <2e-16 ***
restroom   -0.801460   0.006475 -123.78   <2e-16 ***
walleye     0.832039   0.014237   58.44   <2e-16 ***
salmon     -1.256075   0.038518  -32.61   <2e-16 ***
panfish     0.327855   0.002933  111.77   <2e-16 ***
shares_100  0.686990   0.003647  188.37   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.239 on 240394 degrees of freedom
Multiple R-squared:  0.2762,    Adjusted R-squared:  0.2761
F-statistic: 1.529e+04 on 6 and 240394 DF,  p-value: < 2.2e-16
```

Interpretation: In the BLP measurement, the congestion term doesn't have a quantitative impact on the marginal utility estimation of travel cost and interaction terms between individual heterogeneity and site attributes. Regarding the coefficients of site attributes, the magnitudes of marginal utility show a declined trend too and all the estimation is significant. The marginal utility of salmon captures the most dramatic change from 2.221 to -1.256, suggesting the dis-utility of salmon catching. People may prefer other species to salmon, when the congestion level is controlled.

Regarding the effect of congestion, all three model gave a positive measurement of the marginal utility of site shares (around 0.68-0.69), suggesting people prefer crowded fishing sites. The result is actually contrary to our common sense and intuition, and could be attributed to its endogenous property. The congestion level is endogenously determined by the sorting process.

Question 3.ii

As explained in the note session at the beginning, the implementation of quantile IV with GMM is replaced by quantile IV method (proposed by Chernozhukov and Hansen (2006)) and 2SLS. Results comparison is provided below.

2SLS result:

```
Call:
ivreg(formula = data_iv$theta_j ~ data_iv$shares_100 + data_iv$ramp +     data_iv$restroom + data_iv$walleye + data_iv$salmon + data_iv$panfish |     data_iv$shar
e_j + data_iv$ramp + data_iv$restroom + data_iv$walleye +         data_iv$salmon + data_iv$panfish)

Coefficients:
       (Intercept)  data_iv$shares_100        data_iv$ramp    data_iv$restroom     data_iv$walleye      data_iv$salmon     data_iv$panfish
            0.7285             -3.7598             -0.3720             -0.4888              8.1968             20.3990              1.2842
```

Chernozhukov and Hansen (2006) result:

```
Coefficients of endogenous variables:

           tau= 0.5
shares_100     -4.5

Coefficients of exogenous variables:

              tau= 0.5
(Intercept)  0.7827338
ramp        -1.0829497
restroom     0.2425238
walleye     10.1172383
salmon      13.6543978
panfish      1.2730882
```

I would use the result derived by the 2SLS method for welfare analysis in question 4. Because it is found that the quantile IV method is very sensitive to the starting point of estimation. When I was trying several different starting points, the results tend to be very sparse. The current result is achieved when the starting point of $share * 100$ is set to -4.5.

Interpretation based on 2SLS result: The estimation eventually gave a negative result of the marginal utility of shares, suggesting a congestion effect in people's site choosing process instead of an agglomeration effect as shown in the previous models. It is interesting to find out that the coefficient of site attributes increase dramatically especially for the catch rates of walleye and salmon. It suggests a strong correlation between these site attributes and the congestion level, which is omitted in previous models and leads to a downward-biased estimation.

Question 4

Welfare analysis: Partial Equilibrium

- scenario A: 5.217842
- scenario B(affected): 5.09357
- scenario B(unaffected): 2.993604
- scenario C(unaffected by removal): 9.157485e-17
- scenario D(affected): -4.563843
- scenario D(unaffected): -2.341332

Welfare analysis: General Equilibrium

- scenario A: -147.337
- scenario B(for individuals who resort to the affected sites): NaN
- scenario B(unaffected): -144.6079
- scenario C(unaffected by removal): -350.9764
- scenario D(for individuals who resort to the affected sites): NaN
- scenario D(unaffected): -345.9996

Interpretation

- NaN results in general equilibrium suggest that, people don't resort to those sites with higher congestion level, even though the shock increases the walleye catch rates there. The result is even more sensible when the congested sites charge higher entry fee.
- Two measurements vary significantly. That could be due to people's resorting behaviours and the congestion effect. For example, People who went to the congested fishing sites before resort to other sites when there is a policy shock (entry fee/ site attributes improvements), making those new selected sites more crowded, bringing dis-utility.

```
install.packages("mlogit")
install.packages("dplyr")
library(mlogit)
library(dplyr)
setwd("/Volumes/USB30FD/821_ps")
#####data upload and prepare
data <- read.csv("long_data.csv")
data <- data[-c(1)]
names(data)[names(data) == "idcase"] <- "id"
data <- mlogit.data(data, choice = "choice",
                    shape = "long", alt.levels = (c(1:100)), id ="id")
#Q2.i:try the simple model first
model_1 <- mlogit (choice ~ ramp + restroom + walleye + salmon
                   + panfish + travelcost | 0, data)
summary(model_1)

#Q2.ii: preference heterogenerity r.t. person_specific variables
data$panfish_kids <- data$panfish * data$kids
data$restroom_kids <- data$restroom * data$kids
data$ramp_boat <- data$ramp * data$boat
data$walleye_boat <- data$walleye * data$boat

#write.csv(data, file="data_forloop.csv")

model_2 <- mlogit (choice ~ ramp + restroom + walleye + salmon + panfish
                   + travelcost + panfish_kids + restroom_kids
                   + ramp_boat + walleye_boat | 0, data)
summary(model_2)

#Q2.iii: include an unobserved  site attribute
library(mlogit)
library(dplyr)
library(maxLik)
setwd("/Volumes/USB30FD/821_ps")
data <-read.csv("data_forloop.csv")
data <- data[-c(1)]
data$alt_id <-as.numeric(as.character(data$alt_id))
data$id <-as.numeric(as.character(data$id))
##small loop
delta = 1e-07 #tolerancel level for first loop to stop
a <-  c(-0.1030564, -0.1632874, 0.3442013, 0.9818902, 0.6961032)

theta_1 <- replicate(100,0)
theta_2 <- replicate(100,0)
data$theta_j = 0
data$choice <- ifelse(data$choice == "FALSE", 0, 1)
#function that produce the prediction of share S_j for all j
maxlikelihood <- c(2000000000)
b <-c(0)
##
repeat{
  data$Prob_ij_nominator <- exp(0 + a[1]*data$travelcost
+a[2]*data$panfish_kids + a[3]*data$restroom_kids+a[4]*data$ramp_boat +
a[5]*data$walleye_boat)
```

```
  data <- data %>%
    group_by(id) %>%
    mutate(Prob_ij_denom = sum(Prob_ij_nominator))%>%
    ungroup()

  data$Prob_ij <- data$Prob_ij_nominator / data$Prob_ij_denom
  data <- data %>%
    group_by(alt_id) %>%
    mutate(share_j = mean(Prob_ij))%>%
    ungroup()

  share <- data$share_j[1:100]

  real_share <- data$shares[1:100]

  for (j in 1:100) {
    theta_2[j] <- theta_1[j] + log(real_share[j]) - log(share[j])
  }

  theta_diff <- theta_2 - theta_1



  while (max(abs(theta_diff))>delta){
    theta_1 <- theta_2 #update new thetas
    data$theta_j <- theta_1[data$alt_id]
    data <- data %>%
      mutate(Prob_ij_nominator = exp(data$theta_j + a[1]*data$travelcost
+a[2]*data$panfish_kids + a[3]*data$restroom_kids+a[4]*data$ramp_boat +
a[5]*data$walleye_boat))%>%
      group_by(id) %>%
      mutate(Prob_ij_denom = sum(Prob_ij_nominator))%>%
      ungroup()
    data$Prob_ij <- data$Prob_ij_nominator / data$Prob_ij_denom
    data <- data %>%
      group_by(alt_id) %>%
      mutate(share_j = mean(Prob_ij)) %>%
      ungroup()
    share <- data$share_j[1:100]
    for (j in 1:100) {
      theta_2[j] <- theta_1[j] + log(real_share[j]) - log(share[j])
    }
    theta_diff <- theta_2 - theta_1
    #print(theta_2)
  }

  # theta_2 now is the optimised baseline utility for the specific "a"
parameter set.
  # level normalization: substract the mean
  level_norm <- function (x) {
    scale(x, scale = FALSE)
  }
```

```
  baseline_util <- level_norm(theta_2)
  ##update the new theta_j to the dataframe
  data$theta_j <- baseline_util[data$alt_id]

  #x6<- replicate(2404, baseline_util)

  ## calculate likelihood function based on the baseline_utility and
parameter set

  #changed here!!!! be careful
  data <- data %>%
    mutate(Prob_ij_nominator = exp(data$theta_j + a[1]*data$travelcost
+a[2]*data$panfish_kids + a[3]*data$restroom_kids+a[4]*data$ramp_boat +
a[5]*data$walleye_boat))%>%
    group_by(id) %>%
    mutate(Prob_ij_denom = sum(Prob_ij_nominator))%>%
    ungroup()
  data$Prob_ij <- data$Prob_ij_nominator / data$Prob_ij_denom
  #
  data$Prob_sitechoice <- log(data$Prob_ij^data$choice)
  LogL_attributes <- data$Prob_sitechoice
  llmax <- sum(LogL_attributes)
  maxlikelihood[-llmax < maxlikelihood] <- llmax

  ## update parameter set by conducting the maximisation loglikelihood
  #compute loglikelood function  x1, x2, x3, x4, x5, x6

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

  loglike = function(theta, x) {
    # theta parameter vector; x
    mu1 = theta[1]
    mu2 = theta[2]
    mu3 = theta[3]
    mu4 = theta[4]
    mu5 = theta[5]
    x[, 20] <- exp( x[, 19] + mu1*x[, 8] +mu2*x[, 15] + mu3*x[, 16]+mu4*x[,
17] + mu5*x[, 18])
    x1 <- as.vector(unlist(x[,20]))
    x2 <- unname(tapply(x1,(seq_along(x1)-1) %/% 100, sum))
    x[, 21] <- cbind(rep(x2, each=100))
    x[, 22] <- x[, 20] / x[, 21]
    x[, 24]<- x[, 22]^x[, 3]
    ans = sum(log(x[, 24]))
    return(ans)
  }
  theta.start = a
  names(theta.start) = c("mu1", "mu2", "mu3", "mu4", "mu5")
  theta.mle = maxLik (loglike, start=theta.start, x=data, method = "BFGS")
  #theta.mle = optim(par=theta.start, fn=loglike, x=data, method ="BFGS")
  summary(theta.mle)
  summary(theta.mle$coef)
```

```
  # write an additional loop for the first stage to converge
  # print(theta.mle$estimate)
  print(theta.mle$maximum)
  a[theta.mle$maximum > maxlikelihood] <- theta.mle$estimate
  b[round(theta.mle$maximum, 3) == round(maxlikelihood, 3)] <- 1
  maxlikelihood[round(theta.mle$maximum, 3) > round(maxlikelihood, 3)] <-
theta.mle$maximum
  print(a)
  if( b > 0){
    break
  }
}



names(a) <- c("travelcost", "panfish_kids", "restroom_kids", "ramp_boat",
"walleye_boat")

print (maxlikelihood)
print(a)
###BLP second stage
##
BLP2 <- lm(theta_j ~ ramp + restroom + walleye + salmon + panfish + 0,
data=data)
summary(BLP2)

##############################################################################################
##############################################################################################
###Q2.iv
#clean the global environment for the new question and upload package and
settings we need
rm(list=ls())
library(mlogit)
library(dplyr)
library(maxLik)
setwd("C:/821_ps")
data <-read.csv("data_forloop.csv")
data <- data[-c(1)]
data$alt_id <-as.numeric(as.character(data$alt_id))
data$id <-as.numeric(as.character(data$id))
##small loop
delta = 1e-07 #tolerancel level for first loop to stop
theta_1 <- replicate(100,0)
theta_2 <- replicate(100,0)
data$theta_j = 0
data$choice <- ifelse(data$choice == "FALSE", 0, 1)
#function that produce the prediction of share S_j for all j
maxlikelihood <- c(2000000000)
b <-c(0)
a <-  c(-0.1230331, -0.1408312,  0.5068775,  1.3974301,  0.5928853 , 1) ##1
is the initial parameter setting for normal distribution of random
assignment
##add a random component in preferences for walleye
```

```
## and we also take the parameter set from Q2.iii for the initial guess.
The initial guess for delta_j is still zero for all j
## generate random value rnorm(2404, mean=0, sd=a[6])

repeat{
  random <- rnorm(2404, mean=0, sd=a[5])
  data$random <- rep(random, each=100)
  data$Prob_ij_nominator <- exp(0 + a[1]*data$travelcost
+a[2]*data$panfish_kids + a[3]*data$restroom_kids+a[4]*data$ramp_boat +
a[5]*data$walleye_boat + data$random*data$walleye)

  data <- data %>%
    group_by(id) %>%
    mutate(Prob_ij_denom = sum(Prob_ij_nominator))%>%
    ungroup()

  data$Prob_ij <- data$Prob_ij_nominator / data$Prob_ij_denom
  data <- data %>%
    group_by(alt_id) %>%
    mutate(share_j = mean(Prob_ij))%>%
    ungroup()

  share <- data$share_j[1:100]

  real_share <- data$shares[1:100]

  for (j in 1:100) {
    theta_2[j] <- theta_1[j] + log(real_share[j]) - log(share[j])
  }

  theta_diff <- theta_2 - theta_1



  while (max(abs(theta_diff))>delta){
    theta_1 <- theta_2 #update new thetas
    data$theta_j <- theta_1[data$alt_id]
    data <- data %>%
      mutate(Prob_ij_nominator = exp(data$theta_j + a[1]*data$travelcost
+a[2]*data$panfish_kids + a[3]*data$restroom_kids+a[4]*data$ramp_boat +
a[5]*data$walleye_boat + data$random*data$walleye))%>%
      group_by(id) %>%
      mutate(Prob_ij_denom = sum(Prob_ij_nominator))%>%
      ungroup()
    data$Prob_ij <- data$Prob_ij_nominator / data$Prob_ij_denom
    data <- data %>%
      group_by(alt_id) %>%
      mutate(share_j = mean(Prob_ij)) %>%
      ungroup()
    share <- data$share_j[1:100]
    for (j in 1:100) {
      theta_2[j] <- theta_1[j] + log(real_share[j]) - log(share[j])
    }
    theta_diff <- theta_2 - theta_1
```

```
    print(theta_2)
  }

  # theta_2 now is the optimised baseline utility for the specific "a"
parameter set.
  # level normalization: substract the mean
  level_norm <- function (x) {
    scale(x, scale = FALSE)
  }

  baseline_util <- level_norm(theta_2)
  ##update the new theta_j to the dataframe
  data$theta_j <- baseline_util[data$alt_id]

  #x6<- replicate(2404, baseline_util)

  ## calculate likelihood function based on the baseline_utility and
parameter set

  data <- data %>%
    mutate(Prob_ij_nominator = exp(data$theta_j + a[1]*data$travelcost
+a[2]*data$panfish_kids + a[3]*data$restroom_kids+a[4]*data$ramp_boat +
a[5]*data$walleye_boat + data$random*data$walleye))%>%
    group_by(id) %>%
    mutate(Prob_ij_denom = sum(Prob_ij_nominator))%>%
    ungroup()
  data$Prob_ij <- data$Prob_ij_nominator / data$Prob_ij_denom
  #
  data$Prob_sitechoice <- log(data$Prob_ij^data$choice)
  LogL_attributes <- data$Prob_sitechoice
  llmax <- sum(LogL_attributes)
  maxlikelihood[-llmax < maxlikelihood] <- llmax

  ## update parameter set by conducting the maximisation loglikelihood
  #compute loglikelood function  x1, x2, x3, x4, x5, x6

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

  loglike = function(theta, x) {
    # theta parameter vector; x
    mu1 = theta[1]
    mu2 = theta[2]
    mu3 = theta[3]
    mu4 = theta[4]
    mu5 = theta[5]
    x[, 21] <- exp( x[, 19] + mu1*x[, 8] +mu2*x[, 15] + mu3*x[, 16]+mu4*x[,
17] + mu5*x[, 18] + x[,20]*x[,11])
    x1 <- as.vector(unlist(x[,21]))
    x2 <- unname(tapply(x1,(seq_along(x1)-1) %/% 100, sum))
    x[, 22] <- cbind(rep(x2, each=100))
    x[, 23] <- x[, 21] / x[, 22]
    x[, 25]<- x[, 23]^x[, 3]
    ans = sum(log(x[, 25]))
    return(ans)
```

```
  }

  theta.start = a
  names(theta.start) = c("mu1", "mu2", "mu3", "mu4", "mu5", "sigma")
  theta.mle = maxLik (loglike, start=theta.start, x=data, method = "BFGS")
  #theta.mle = optim(par=theta.start, fn=loglike, x=data, method ="BFGS")
  summary(theta.mle)
  summary(theta.mle$coef)


  # write an additional loop for the first stage to converge
  # print(theta.mle$estimate)
  print(theta.mle$maximum)
  a[theta.mle$maximum > maxlikelihood] <- theta.mle$estimate
  a[6] <- sd(data$random)
  b[round(theta.mle$maximum, 3) == round(maxlikelihood, 3)] <- 1
  maxlikelihood[theta.mle$maximum > maxlikelihood] <- theta.mle$maximum
  print(a)
  if( b > 0){
    break
  }
}



print(a)
print (llmax)
print(theta.mle$estimate)



###BLP second stage
##
BLP2 <- lm(theta_j ~ ramp + restroom + walleye + salmon + panfish + 0,
data=data)
summary(BLP2)

###################Q3
library(mlogit)
library(dplyr)
library(maxLik)
setwd("/Volumes/USB30FD/821_ps")
#####data upload and prepare
data <- read.csv("long_data.csv")
data <- data[-c(1)]
names(data)[names(data) == "idcase"] <- "id"
data <- mlogit.data(data, choice = "choice",
                    shape = "long", alt.levels = (c(1:100)), id ="id")
data$shares <- data$shares*100
names(data)[names(data) == "shares"] <- "shares_100"
#Q3.1i:try the simple model first
model_1 <- mlogit (choice ~ ramp + restroom + walleye + salmon
                   + panfish + travelcost + shares_100| 0, data)
```

```
summary(model_1)


#Q3.1ii: preference heterogenerity r.t. person_specific variables
data$panfish_kids <- data$panfish * data$kids
data$restroom_kids <- data$restroom * data$kids
data$ramp_boat <- data$ramp * data$boat
data$walleye_boat <- data$walleye * data$boat

model_2 <- mlogit (choice ~ ramp + restroom + walleye + salmon + panfish
                   + travelcost + panfish_kids + restroom_kids
                   + ramp_boat + walleye_boat + shares_100 | 0, data)
summary(model_2)



#Q3.1iii: BLP
data$alt_id <-as.numeric(as.character(data$alt_id))
data$id <-as.numeric(as.character(data$id))
##small loop
delta = 1e-07 #tolerancel level for first loop to stop
a <-  c(0, 0, 0, 0, 0)

theta_1 <- replicate(100,0)
theta_2 <- replicate(100,0)
data$theta_j = 0
data$choice <- ifelse(data$choice == "FALSE", 0, 1)
#function that produce the prediction of share S_j for all j
maxlikelihood <- c(2000000000)
b <-c(0)
##
repeat{
  data$Prob_ij_nominator <- exp(0 + a[1]*data$travelcost
+a[2]*data$panfish_kids + a[3]*data$restroom_kids+a[4]*data$ramp_boat +
a[5]*data$walleye_boat)

  data <- data %>%
    group_by(id) %>%
    mutate(Prob_ij_denom = sum(Prob_ij_nominator))%>%
    ungroup()

  data$Prob_ij <- data$Prob_ij_nominator / data$Prob_ij_denom
  data <- data %>%
    group_by(alt_id) %>%
    mutate(share_j = mean(Prob_ij)*100)%>%
    ungroup()

  share <- data$share_j[1:100]

  real_share <- data$shares_100[1:100]

  for (j in 1:100) {
    theta_2[j] <- theta_1[j] + log(real_share[j]) - log(share[j])
  }
```

```
  theta_diff <- theta_2 - theta_1



  while (max(abs(theta_diff))>delta){
    theta_1 <- theta_2 #update new thetas
    data$theta_j <- theta_1[data$alt_id]
    data <- data %>%
      mutate(Prob_ij_nominator = exp(data$theta_j + a[1]*data$travelcost
+a[2]*data$panfish_kids + a[3]*data$restroom_kids+a[4]*data$ramp_boat +
a[5]*data$walleye_boat))%>%
      group_by(id) %>%
      mutate(Prob_ij_denom = sum(Prob_ij_nominator))%>%
      ungroup()
    data$Prob_ij <- data$Prob_ij_nominator / data$Prob_ij_denom
    data <- data %>%
      group_by(alt_id) %>%
      mutate(share_j = mean(Prob_ij)*100) %>%
      ungroup()
    share <- data$share_j[1:100]
    for (j in 1:100) {
      theta_2[j] <- theta_1[j] + log(real_share[j]) - log(share[j])
    }
    theta_diff <- theta_2 - theta_1
    #print(theta_2)
  }

  # theta_2 now is the optimised baseline utility for the specific "a"
parameter set.
  # level normalization: substract the mean
  level_norm <- function (x) {
    scale(x, scale = FALSE)
  }

  baseline_util <- level_norm(theta_2)
  ##update the new theta_j to the dataframe
  data$theta_j <- baseline_util[data$alt_id]

  #x6<- replicate(2404, baseline_util)

  ## calculate likelihood function based on the baseline_utility and
parameter set

  #changed here!!!! be careful
  data <- data %>%
    mutate(Prob_ij_nominator = exp(data$theta_j + a[1]*data$travelcost
+a[2]*data$panfish_kids + a[3]*data$restroom_kids+a[4]*data$ramp_boat +
a[5]*data$walleye_boat))%>%
    group_by(id) %>%
    mutate(Prob_ij_denom = sum(Prob_ij_nominator))%>%
    ungroup()
  data$Prob_ij <- data$Prob_ij_nominator / data$Prob_ij_denom
  #
```

```
  data$Prob_sitechoice <- log(data$Prob_ij^data$choice)
  LogL_attributes <- data$Prob_sitechoice
  llmax <- sum(LogL_attributes)
  maxlikelihood[-llmax < maxlikelihood] <- llmax

  ## update parameter set by conducting the maximisation loglikelihood
  #compute loglikelood function  x1, x2, x3, x4, x5, x6

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

  loglike = function(theta, x) {
    # theta parameter vector; x
    mu1 = theta[1]
    mu2 = theta[2]
    mu3 = theta[3]
    mu4 = theta[4]
    mu5 = theta[5]
    x[, 20] <- exp( x[, 19] + mu1*x[, 8] +mu2*x[, 15] + mu3*x[, 16]+mu4*x[,
17] + mu5*x[, 18])
    x1 <- as.vector(unlist(x[,20]))
    x2 <- unname(tapply(x1,(seq_along(x1)-1) %/% 100, sum))
    x[, 21] <- cbind(rep(x2, each=100))
    x[, 22] <- x[, 20] / x[, 21]
    x[, 24]<- x[, 22]^x[, 3]
    ans = sum(log(x[, 24]))
    return(ans)
  }
  theta.start = a
  names(theta.start) = c("mu1", "mu2", "mu3", "mu4", "mu5")
  theta.mle = maxLik (loglike, start=theta.start, x=data, method = "BFGS")
  #theta.mle = optim(par=theta.start, fn=loglike, x=data, method ="BFGS")
  summary(theta.mle)
  summary(theta.mle$coef)


  # write an additional loop for the first stage to converge
  # print(theta.mle$estimate)
  print(theta.mle$maximum)
  a[theta.mle$maximum > maxlikelihood] <- theta.mle$estimate
  b[round(theta.mle$maximum, 3) == round(maxlikelihood, 3)] <- 1
  maxlikelihood[round(theta.mle$maximum, 3) > round(maxlikelihood, 3)] <-
theta.mle$maximum
  print(a)
  if( b > 0){
    break
  }
}


names(a) <- c("travelcost", "panfish_kids", "restroom_kids", "ramp_boat",
"walleye_boat")
print(a)
print (maxlikelihood)
print(theta.mle$estimate)
```

```
##> print(a)
##[1] -0.1230306 -0.  1408963  0.5048161  1.3881009  0.5524893
##> print (maxlikelihood)
##[1] -5136.971
##> print(theta.mle$estimate)
##mu1        mu2        mu3        mu4        mu5
##-0.1230306 -0.1408963  0.5048161  1.3881009  0.5524893

###BLP second stage
##
BLP2 <- lm(theta_j ~ ramp + restroom + walleye + salmon + panfish +
shares_100 + 0, data=data)
summary(BLP2)
#############################################################################################

#Q3.2
library(quantreg)
library(gmm)
###arange data frame
data_iv <- data[1:100 ,]
data_iv <- subset(data_iv, select=c("ramp", "restroom",
"walleye","salmon","panfish","shares_100","theta_j"))
############generate instruments
###median regression
rqfit <- rq(theta_j ~ ramp + restroom + walleye + salmon + panfish +
shares_100, data=data_iv)
coef <-rqfit$coefficients
###calculate shares as one of instruments
data$Prob_ij_nominator <- exp(coef[1] + coef[2]*data$ramp +
coef[3]*data$restroom + coef[4]*data$walleye + coef[5]*data$salmon +
coef[6]*data$panfish + a[1]*data$travelcost +a[2]*data$panfish_kids +
a[3]*data$restroom_kids+a[4]*data$ramp_boat + a[5]*data$walleye_boat)

data <- data %>%
  group_by(id) %>%
  mutate(Prob_ij_denom = sum(Prob_ij_nominator))%>%
  ungroup()

data$Prob_ij <- data$Prob_ij_nominator / data$Prob_ij_denom
data <- data %>%
  group_by(alt_id) %>%
  mutate(share_j = mean(Prob_ij))%>%
  ungroup()

data_iv <- data[1:100 ,]
data_iv <- subset(data_iv, select=c("ramp", "restroom",
"walleye","salmon","panfish","shares_100","theta_j","share_j"))
data_iv$share_j <- data_iv$share_j*100
####quantile IV GMM
##generate condition function
g1 <- function (tet, x){
  #tet <- parameter set
  #x <- data_iv dataframe
```

```
  #Sn <- 0.25, same as the setting of orginal paper
  # intercept from median regression <- -0.1190958
  m1 <- (pnorm((x[,7] + 0.1190958 - tet[1]*x[,1] - tet[2]*x[,2] -
tet[3]*x[,3] - tet[4]*x[,4] - tet[5]*x[,5] - tet[6]*x[,8])/0.25) - 0.5 )
  return(m1)
}

####other method: "inverse" quantile estimation
install.packages("remotes")
remotes::install_github("yuchang0321/IVQR")
library(IVQR)

fit <- ivqr(theta_j~ shares_100 | share_j | ramp + restroom + walleye +
salmon + panfish, 0.5, grid= seq(-4.5 , 0 , 0.05625), data = data_iv)
fit
####Other Method: 2sls
#OLS
ols<- lm(theta_j ~ ramp + restroom + walleye + salmon + panfish +
shares_100 + 0, data=data_iv)
coef <-ols$coefficients
#predict shares of visiting as instrument, based on exogenous things only
data$Prob_ij_nominator <- exp( coef[1]*data$ramp + coef[2]*data$restroom +
coef[3]*data$walleye + coef[4]*data$salmon + coef[5]*data$panfish +
a[1]*data$travelcost +a[2]*data$panfish_kids +
a[3]*data$restroom_kids+a[4]*data$ramp_boat + a[5]*data$walleye_boat)

data <- data %>%
  group_by(id) %>%
  mutate(Prob_ij_denom = sum(Prob_ij_nominator))%>%
  ungroup()

data$Prob_ij <- data$Prob_ij_nominator / data$Prob_ij_denom
data <- data %>%
  group_by(alt_id) %>%
  mutate(share_j = mean(Prob_ij))%>%
  ungroup()

data_iv <- data[1:100 ,]
data_iv <- subset(data_iv, select=c("ramp", "restroom",
"walleye","salmon","panfish","shares_100","theta_j","share_j"))
data_iv$share_j <- data_iv$share_j*100

#2SLS
library(ivpack)
twosls <- ivreg(data_iv$theta_j~ data_iv$shares_100 + data_iv$ramp +
data_iv$restroom +data_iv$walleye +data_iv$salmon + data_iv$panfish |
data_iv$share_j + data_iv$ramp + data_iv$restroom +data_iv$walleye
+data_iv$salmon + data_iv$panfish   )
beta <- c(-3.7598, -0.3720, -0.4888, 8.1968, 20.3990, 1.2842) ##from Q3_2
names(beta) <- c("shares_100" , "ramp", "restroom", "walleye", "salmon",
"panfish")
data_iv$unobserved <- data_iv$theta_j - beta[1]*data_iv$shares_100 -
beta[2]*data_iv$ramp - beta[3]*data_iv$restroom -beta[4]*data_iv$walleye -
beta[5]*data_iv$salmon - beta[6]*data_iv$panfish
```

```
####################################################
##data preparation for Q4
unobserved <- data_iv$unobserved

data <- read.csv("long_data.csv")
data <- data[-c(1)]
names(data)[names(data) == "idcase"] <- "id"
data <- mlogit.data(data, choice = "choice",
                    shape = "long", alt.levels = (c(1:100)), id ="id")
data$shares <- data$shares*100
names(data)[names(data) == "shares"] <- "shares_100"
data$panfish_kids <- data$panfish * data$kids
data$restroom_kids <- data$restroom * data$kids
data$ramp_boat <- data$ramp * data$boat
data$walleye_boat <- data$walleye * data$boat
data$alt_id <-as.numeric(as.character(data$alt_id))
data$id <-as.numeric(as.character(data$id))
data$unobserved <- cbind(rep( unobserved, 2404))
write.csv(data, file = "data_for_welfare_analysis.csv")

###### Welfare Analysis
##data
library(dplyr)
setwd("C:/821_ps")
data <- read.csv("data_for_welfare_analysis.csv")
data <- data[-c(1)]
data$id <-as.numeric(as.character(data$id))

##generate utility calculation function
beta <- c(-3.7598, -0.3720, -0.4888, 8.1968, 20.3990, 1.2842) ##from Q3_2
names(beta) <- c("shares_100" , "ramp", "restroom", "walleye", "salmon",
"panfish")
alpha <- c(-0.1230306, -0.1408963,  0.5048161,  1.3881009,  0.5524893 )
names(alpha) <- c("travelcost", "panfish_kids", "restroom_kids",
"ramp_boat", "walleye_boat")
U <- function(beta, alpha, x) {
  ans <- beta[1]*x[,14] + beta[2]*x[,9] + beta[3]*x[,10]+ beta[4]*x[,11] +
beta[5]*x[,12] + beta[6]*x[,13] + alpha[1]*x[,8] + alpha[2]*x[,15] +
alpha[3]*x[,16] + alpha[4]*x[,17] + alpha[5]*x[,18]+ x[,19]
  return(ans)

}
#####################i Partial equilibrium
data$old_utility <- U(beta, alpha, data)
write.csv(data, file="data_welfare.csv")
### scenario A
data$walleye <- data$walleye*1.3
data$walleye_boat <-data$walleye*data$boat
data$new_utility <- U(beta, alpha, data)

data$old_utility <- exp(data$old_utility)
data$new_utility <- exp(data$new_utility)

new_utility <- data$new_utility
```

```
sum_new <- unname(tapply(new_utility,(seq_along(new_utility)-1) %/% 100,
sum))
data$sum_new <- cbind(rep(sum_new, each =100))
data$sum_new <- log(data$sum_new)

old_utility <- data$old_utility
sum_old <- unname(tapply(old_utility,(seq_along(old_utility)-1) %/% 100,
sum))
data$sum_old <- cbind(rep(sum_old, each =100))
data$sum_old <- log(data$sum_old)

data$CV <- (1/0.123131)*(data$sum_new-data$sum_old)
mean(data$CV)   ###  5.217842


### scenario B
data <- read.csv("data_welfare.csv")
data <- data[-c(1)]
data$change <- ifelse(data$shares_100 >1.5, 1.3, 1)
data$walleye <- data$walleye*data$change
data$walleye_boat <-data$walleye*data$boat
data$new_utility <- U(beta, alpha, data)

data$old_utility <- exp(data$old_utility)
data$new_utility <- exp(data$new_utility)

new_utility <- data$new_utility
sum_new <- unname(tapply(new_utility,(seq_along(new_utility)-1) %/% 100,
sum))
data$sum_new <- cbind(rep(sum_new, each =100))
data$sum_new <- log(data$sum_new)

old_utility <- data$old_utility
sum_old <- unname(tapply(old_utility,(seq_along(old_utility)-1) %/% 100,
sum))
data$sum_old <- cbind(rep(sum_old, each =100))
data$sum_old <- log(data$sum_old)
data$CV <- (1/0.123131)*(data$sum_new-data$sum_old)

mean_affected <- data$CV[which(data$change == 1.3 & data$choice =="TRUE") ]
%>% mean() ###5.09357
mean_unaffected <- mean(data$CV[which(data$change == 1.0 & data$choice
=="TRUE") ]) %>% mean() ###2.993604


### scenario C
data <- read.csv("data_welfare.csv")
data <- data[-c(1)]
data <- data[which(data$shares_100 <= 1.5),] ##now we have 82 sites

data$new_utility <- U(beta, alpha, data)

data$old_utility <- exp(data$old_utility)
data$new_utility <- exp(data$new_utility)
```

```
new_utility <- data$new_utility
sum_new <- unname(tapply(new_utility,(seq_along(new_utility)-1) %/% 82,
sum))
data$sum_new <- cbind(rep(sum_new, each =82))
data$sum_new <- log(data$sum_new)

old_utility <- data$old_utility
sum_old <- unname(tapply(old_utility,(seq_along(old_utility)-1) %/% 82,
sum))
data$sum_old <- cbind(rep(sum_old, each =82))
data$sum_old <- log(data$sum_old)
data$CV <- (1/0.123131)*(data$sum_new-data$sum_old)

mean_unaffected_by_removal <- data$CV[which(data$choice =="TRUE") ] %>%
mean() ###9.157485e-17

### scenario D
data <- read.csv("data_welfare.csv")
data <- data[-c(1)]
data$change <- ifelse(data$shares_100 >1.5, 10, 0)
data$travelcost <- data$travelcost+data$change

data$new_utility <- U(beta, alpha, data)

data$old_utility <- exp(data$old_utility)
data$new_utility <- exp(data$new_utility)

new_utility <- data$new_utility
sum_new <- unname(tapply(new_utility,(seq_along(new_utility)-1) %/% 100,
sum))
data$sum_new <- cbind(rep(sum_new, each =100))
data$sum_new <- log(data$sum_new)

old_utility <- data$old_utility
sum_old <- unname(tapply(old_utility,(seq_along(old_utility)-1) %/% 100,
sum))
data$sum_old <- cbind(rep(sum_old, each =100))
data$sum_old <- log(data$sum_old)
data$CV <- (1/0.123131)*(data$sum_new-data$sum_old)

mean_affected <- data$CV[which(data$change == 10 & data$choice =="TRUE") ]
%>% mean() ###-4.563843
mean_unaffected <- mean(data$CV[which(data$change == 0 & data$choice
=="TRUE") ]) %>% mean() ###-2.341332


######################ii general equilibrium <- resorting

### scenario A
data <- read.csv("data_welfare.csv")
data <- data[-c(1)]
data$walleye <- data$walleye*1.3
data$walleye_boat <-data$walleye*data$boat
```

```
#stimulate new choice set
data$nominator <- exp(U(beta, alpha, data))
nominator <- data$nominator
denominator <- unname(tapply(nominator,(seq_along(nominator)-1) %/% 100,
sum))
data$denomitor <- cbind(rep(denominator, each =100))
data$pij <- data$nominator/data$denomitor
data<-arrange(data, data$alt_id)
pij <- data$pij
pij_mean <- unname(tapply(pij,(seq_along(pij)-1) %/% 2404, sum))
data$shares_new <- cbind(rep(pij_mean, each =2404))
data<-arrange(data, data$id)
##calculate new utility
U_GE <- function(beta, alpha, x) {
  ans <- beta[1]*x[,24] + beta[2]*x[,9] + beta[3]*x[,10]+ beta[4]*x[,11] +
beta[5]*x[,12] + beta[6]*x[,13] + alpha[1]*x[,8] + alpha[2]*x[,15] +
alpha[3]*x[,16] + alpha[4]*x[,17] + alpha[5]*x[,18]+ x[,19]
  return(ans)
}
data$new_utility <- U_GE(beta, alpha, data)

data$old_utility <- exp(data$old_utility)
data$new_utility <- exp(data$new_utility)

new_utility <- data$new_utility
sum_new <- unname(tapply(new_utility,(seq_along(new_utility)-1) %/% 100,
sum))
data$sum_new <- cbind(rep(sum_new, each =100))
data$sum_new <- log(data$sum_new)

old_utility <- data$old_utility
sum_old <- unname(tapply(old_utility,(seq_along(old_utility)-1) %/% 100,
sum))
data$sum_old <- cbind(rep(sum_old, each =100))
data$sum_old <- log(data$sum_old)

data$CV <- (1/0.123131)*(data$sum_new-data$sum_old)
mean(data$CV)   ###  -147.337

### scenario B
data <- read.csv("data_welfare.csv")
data <- data[-c(1)]
data$change <- ifelse(data$shares_100 >1.5, 1.3, 1)
data$walleye <- data$walleye*data$change
data$walleye_boat <-data$walleye*data$boat
#stimulate new choice set
data$nominator <- exp(U(beta, alpha, data))
nominator <- data$nominator
denominator <- unname(tapply(nominator,(seq_along(nominator)-1) %/% 100,
sum))
data$denomitor <- cbind(rep(denominator, each =100))
data$pij <- data$nominator/data$denomitor
data<-arrange(data, data$alt_id)
pij <- data$pij
```

```
pij_mean <- unname(tapply(pij,(seq_along(pij)-1) %/% 2404, sum))
data$shares_new <- cbind(rep(pij_mean, each =2404))
data<-arrange(data, data$id)
##calculate new utility
U_GE <- function(beta, alpha, x) {
  ans <- beta[1]*x[,25] + beta[2]*x[,9] + beta[3]*x[,10]+ beta[4]*x[,11] +
beta[5]*x[,12] + beta[6]*x[,13] + alpha[1]*x[,8] + alpha[2]*x[,15] +
alpha[3]*x[,16] + alpha[4]*x[,17] + alpha[5]*x[,18]+ x[,19]
  return(ans)
}
data$new_utility <- U_GE(beta, alpha, data)

data$old_utility <- exp(data$old_utility)
data$new_utility <- exp(data$new_utility)

new_utility <- data$new_utility
sum_new <- unname(tapply(new_utility,(seq_along(new_utility)-1) %/% 100,
sum))
data$sum_new <- cbind(rep(sum_new, each =100))
data$sum_new <- log(data$sum_new)

old_utility <- data$old_utility
sum_old <- unname(tapply(old_utility,(seq_along(old_utility)-1) %/% 100,
sum))
data$sum_old <- cbind(rep(sum_old, each =100))
data$sum_old <- log(data$sum_old)
data$CV <- (1/0.123131)*(data$sum_new-data$sum_old)
## update actual choice
data <- arrange(data, id)
new_utility <- c(data$new_utility)
data$new_utility <- new_utility
data<- arrange(data, id, desc(data$new_utility))
data$new_choice = "FALSE"
for (i in 0:2403 ){
  data$new_choice[(100*i+1)] <- "TRUE"
}

mean_affected <- data$CV[which(data$change == 1.3 & data$new_choice
=="TRUE") ] %>% mean() ###NaN, suggesting no person change to the place
when walleye increased in the crowded place.
mean_unaffected <- mean(data$CV[which(data$change == 1.0 & data$new_choice
=="TRUE") ]) %>% mean() ###-144.6079


###Scenerio C
data <- read.csv("data_welfare.csv")
data <- data[-c(1)]
data <- data[which(data$shares_100 <= 1.5),] ##now we have 82 sites
#stimulate new choice set
data$nominator <- exp(U(beta, alpha, data))
nominator <- data$nominator
denominator <- unname(tapply(nominator,(seq_along(nominator)-1) %/% 82,
sum))
data$denomitor <- cbind(rep(denominator, each =82))
```

```
data$pij <- data$nominator/data$denomitor
data<-arrange(data, data$alt_id)
pij <- data$pij
pij_mean <- unname(tapply(pij,(seq_along(pij)-1) %/% 2404, sum))
data$shares_new <- cbind(rep(pij_mean, each =2404))
data<-arrange(data, data$id)
##calculate new utility
U_GE <- function(beta, alpha, x) {
  ans <- beta[1]*x[,24] + beta[2]*x[,9] + beta[3]*x[,10]+ beta[4]*x[,11] +
beta[5]*x[,12] + beta[6]*x[,13] + alpha[1]*x[,8] + alpha[2]*x[,15] +
alpha[3]*x[,16] + alpha[4]*x[,17] + alpha[5]*x[,18]+ x[,19]
  return(ans)
}
data$new_utility <- U_GE(beta, alpha, data)

data$old_utility <- exp(data$old_utility)
data$new_utility <- exp(data$new_utility)

new_utility <- data$new_utility
sum_new <- unname(tapply(new_utility,(seq_along(new_utility)-1) %/% 82,
sum))
data$sum_new <- cbind(rep(sum_new, each =82))
data$sum_new <- log(data$sum_new)

old_utility <- data$old_utility
sum_old <- unname(tapply(old_utility,(seq_along(old_utility)-1) %/% 82,
sum))
data$sum_old <- cbind(rep(sum_old, each =82))
data$sum_old <- log(data$sum_old)
data$CV <- (1/0.123131)*(data$sum_new-data$sum_old)

mean_unaffected_by_removal <- data$CV[which(data$choice =="TRUE") ] %>%
mean() ###-350.9764

### scenario D
data <- read.csv("data_welfare.csv")
data <- data[-c(1)]
data$change <- ifelse(data$shares_100 >1.5, 10, 0)
data$travelcost <- data$travelcost+data$change
#stimulate new choice set
data$nominator <- exp(U(beta, alpha, data))
nominator <- data$nominator
denominator <- unname(tapply(nominator,(seq_along(nominator)-1) %/% 100,
sum))
data$denomitor <- cbind(rep(denominator, each =100))
data$pij <- data$nominator/data$denomitor
data<-arrange(data, data$alt_id)
pij <- data$pij
pij_mean <- unname(tapply(pij,(seq_along(pij)-1) %/% 2404, sum))
data$shares_new <- cbind(rep(pij_mean, each =2404))
data<-arrange(data, data$id)
##calculate new utility
U_GE <- function(beta, alpha, x) {
```

```
  ans <- beta[1]*x[,25] + beta[2]*x[,9] + beta[3]*x[,10]+ beta[4]*x[,11] +
beta[5]*x[,12] + beta[6]*x[,13] + alpha[1]*x[,8] + alpha[2]*x[,15] +
alpha[3]*x[,16] + alpha[4]*x[,17] + alpha[5]*x[,18]+ x[,19]
  return(ans)
}
data$new_utility <- U_GE(beta, alpha, data)

data$old_utility <- exp(data$old_utility)
data$new_utility <- exp(data$new_utility)

new_utility <- data$new_utility
sum_new <- unname(tapply(new_utility,(seq_along(new_utility)-1) %/% 100,
sum))
data$sum_new <- cbind(rep(sum_new, each =100))
data$sum_new <- log(data$sum_new)

old_utility <- data$old_utility
sum_old <- unname(tapply(old_utility,(seq_along(old_utility)-1) %/% 100,
sum))
data$sum_old <- cbind(rep(sum_old, each =100))
data$sum_old <- log(data$sum_old)
data$CV <- (1/0.123131)*(data$sum_new-data$sum_old)
## update actual choice
data <- arrange(data, id)
new_utility <- c(data$new_utility)
data$new_utility <- new_utility
data<- arrange(data, id, desc(data$new_utility))
data$new_choice = "FALSE"
for (i in 0:2403 ){
  data$new_choice[(100*i+1)] <- "TRUE"
}

mean_affected <- data$CV[which(data$change == 10 & data$new_choice
=="TRUE") ] %>% mean() ###NaN
mean_unaffected <- mean(data$CV[which(data$change == 0 & data$new_choice
=="TRUE") ]) %>% mean() ###-345.9996
```