Lecture note for Discrete Choice Models #3

At
Advanced Transportation Planning

Tetsuro HYODO

"Blue Bus, Red Bus" problem:

There are two modes as follows;

$P_{rail} = \frac{e^{V_{rail}}}{e^{V_{rail}} + e^{V_{bus}}}$, here we assume $V_{rail} = V_{bus}$, so $P_{rail} = P_{bus} = 0.5$

Up to now, the color of buses is blue, and the city transportation sector decided to change the half of bus color to red, so bus has two alternatives such as;

$$P_{rail} = \frac{e^{V_{rail}}}{e^{V_{rail}} + e^{V_{blue-bus}} + e^{V_{red-bus}}}$$

So the share of blue & red bus increases to 2/3.

→ This process does not consider the "similarity among alternatives".

$$\frac{P_{blue-bus}}{P_{red-bus}} = \frac{e^{V_{blue-bus}}}{e^{V_{red-bus}}} = e^{V_{blue-bus} - V_{red-bus}}$$

→ The ratio of blue bus & red bus probability is independent from other alternatives, V_{rail}

→ This property is called "Independence from Irrelevant Alternatives (IIA)"

→ 「選択肢の文脈独立」

→ Sometimes, this property is regarded as one of the disadvantage of Multinomial Logit Model (MNL), so Daniel McFadden developed Nested Logit (NL) Model !

Nested Logit Model (NL):

“Blue bus” vs “Red bus” problem…

→ Independence from Irrelevant Alternatives (IIA)

Let’s generate correlated two normally random variables:

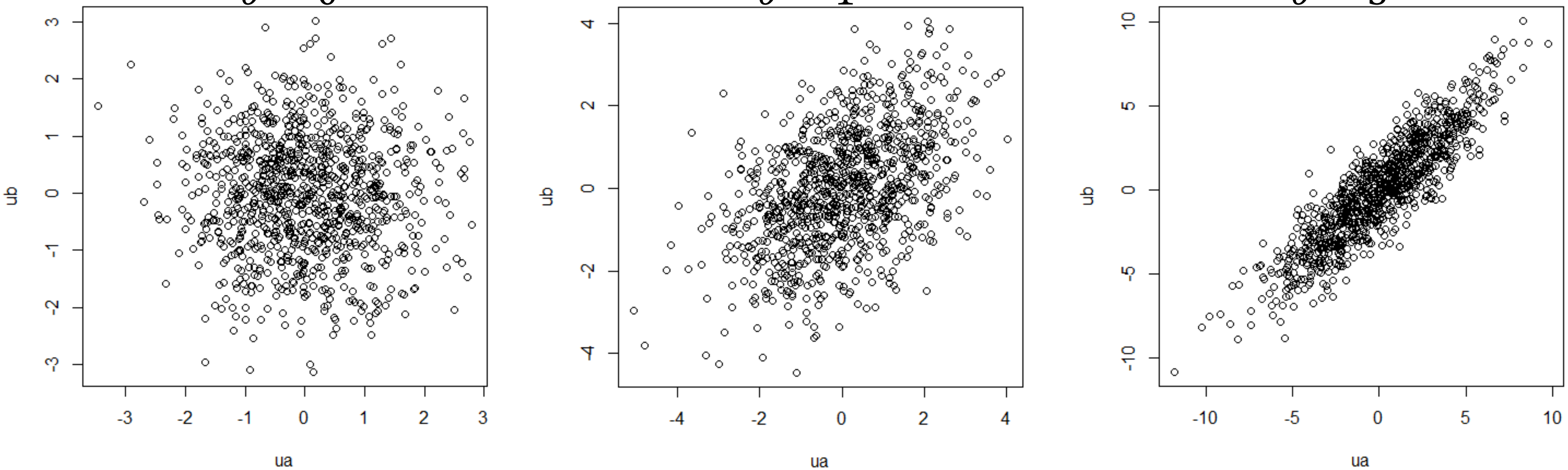
$u_a = \varepsilon_a + \theta\varepsilon_{ab}$ （↓独立同一分布）

$u_b = \varepsilon_b + \theta\varepsilon_{ab}$, $\varepsilon_a, \varepsilon_b, \varepsilon_{ab} \sim N(0,1)$ I.I.D. → independent and identically distributed

$u_a, u_b \sim N(0, 1+\theta^2)$ and covariance of u_a and u_b is θ^2.

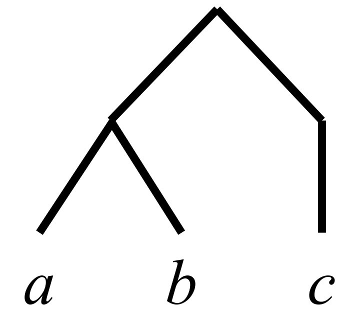
So the coefficient of correlation is $\rho = \frac{\theta^2}{1+\theta^2}$.

$\rho = 0$, $\theta = 0$ $\rho = 0.5$, $\theta = 1$ $\rho = 0.9$, $\theta = 3$




```
a <- rnorm(1000); b <- rnorm(1000); ab <- rnorm(1000); par(ask=T)
rho <- 0  ; theta <- sqrt(rho/(1-rho)); ua <- a + theta*ab; ub <- b + theta*ab; plot(ua,ub)
rho <- 0.5; theta <- sqrt(rho/(1-rho)); ua <- a + theta*ab; ub <- b + theta*ab; plot(ua,ub)
rho <- 0.9; theta <- sqrt(rho/(1-rho)); ua <- a + theta*ab; ub <- b + theta*ab; plot(ua,ub)
```

"Blue bus" vs "Red bus" problem seems "Similarity" problem.
And "Similarity" can be expressed as "Correlation of error terms"



$U_a = V_{a|ab} + \varepsilon_a + \varepsilon_{ab}$
$U_b = V_{b|ab} + \varepsilon_b + \varepsilon_{ab}$
$U_c = V_c + \varepsilon_c + \varepsilon'_c$
$\varepsilon_a, \varepsilon_b, \varepsilon_{ab} \sim G(0, \lambda)$ and $\varepsilon_a + \varepsilon_{ab}, \varepsilon_b + \varepsilon_{ab}, \varepsilon_c + \varepsilon'_c \sim G(0, \lambda')$

$$U_{a|ab} = V_{a|ab} + \varepsilon_a, U_{b|ab} = V_{b|ab} + \varepsilon_b \rightarrow P_{a|ab} = \frac{e^{\lambda V_{a|ab}}}{e^{\lambda V_{a|ab}} + e^{\lambda V_{b|ab}}}$$

And the expectation of maximum value of these two utilities is:

$$U'_{ab} = E\left[max\left(U_{a|ab,} U_{b|ab,}\right)\right] \sim G\left(\frac{1}{\lambda} ln\left(e^{\lambda V_{a|ab}} + e^{\lambda V_{b|ab}}\right), \lambda\right)$$

The upper tree's utilities are:

$$U_{ab} = U'_{ab} + \varepsilon_{ab} \sim G\left(\frac{1}{\lambda} ln\left(e^{\lambda V_{a|ab}} + e^{\lambda V_{b|ab}}\right), \lambda'\right)$$
$$U_c \sim G(V_c, \lambda')$$

So, the probability of a is as follows:

$$P_a = P_{ab} \times P_{a|ab} = \frac{e^{\frac{\lambda'}{\lambda} ln\left(e^{\lambda V_{a|ab}} + e^{\lambda V_{b|ab}}\right)}}{e^{\frac{\lambda'}{\lambda} ln\left(e^{\lambda V_{a|ab}} + e^{\lambda V_{b|ab}}\right)} + e^{\lambda' V_c}} \times \frac{e^{\lambda V_{a|ab}}}{e^{\lambda V_{a|ab}} + e^{\lambda V_{b|ab}}}$$

Parameters and one lambda (λ or λ') can not be estimated separately, here we set $\lambda = 1$, then…

$$P_a = P_{ab} \times P_{a|ab} = \frac{e^{\lambda' ln\left(e^{V_{a|ab}} + e^{V_{b|ab}}\right)}}{e^{\lambda' ln\left(e^{V_{a|ab}} + e^{V_{b|ab}}\right)} + e^{\lambda' V_c}} \times \frac{e^{V_{a|ab}}}{e^{V_{a|ab}} + e^{V_{b|ab}}}$$

λ' is called "logsum" parameter, and $G(0, \lambda')$ is summation of two error terms, so the variance of $G(0, \lambda')$ is greater than the variance of $G(0, \lambda)$.

It means that $\lambda' < \lambda$, therefore, $0 < \frac{\lambda'}{\lambda} = \lambda' < 1$. If λ'=1, what ?

The estimated logsum parameter should satisfy $0 < \lambda' < 1$, if the nested tree structure may be reasonable. In other words, this parameter is the index of model validity.

$$var[U_{ab}] = \frac{\pi^2}{6\lambda'^2}, var\left[U_{a|ab}\right] = \frac{\pi^2}{6}$$

$$\lambda'^2 = \frac{var\left[U_{a|ab}\right]}{var[U_{ab}]} = \frac{var\left[U_{a|ab}\right]}{var\left[U_{a|ab}\right] + var[\varepsilon_{ab}]} = 1 - \frac{var[\varepsilon_{ab}]}{var\left[U_{a|ab}\right] + var[\varepsilon_{ab}]}$$

The common term of error ε_{ab} is covariance. So,

$$\lambda'^2 = 1 - \frac{var[\varepsilon_{ab}]}{var\left[U_{a|ab}\right] + var[\varepsilon_{ab}]} = 1 - \frac{cov\left[U_{a|ab} + \varepsilon_{ab}, U_{b|ab} + \varepsilon_{ab}\right]}{\sqrt{var\left[U_{a|ab} + \varepsilon_{ab}\right] \times var\left[U_{b|ab} + \varepsilon_{ab}\right]}}$$

$$= 1 - \rho$$

By using logsum parameter, correlation of errors can be calculated as $\rho = 1 - \lambda'^2$

From the result of previous page, variance-covariance matrix of three alternatives is

$$\frac{\pi^2}{6}\begin{bmatrix} 1 & 1-\frac{1}{\lambda'^2} & 0 \\ 1-\frac{1}{\lambda'^2} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \frac{\pi^2}{6}\begin{bmatrix} 1 & \rho & 0 \\ \rho & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Multinomial Probit (MNP) model:

Similar error component can be expressed by Multinomial Probit model.

The covariance matrix is defined as follows:

$$\begin{bmatrix} \sigma_a^2 & \sigma_{ab}^2 & 0 \\ \sigma_{ab}^2 & \sigma_b^2 & 0 \\ 0 & 0 & \sigma_c^2 \end{bmatrix} = \sigma^2 \begin{bmatrix} 1 & \rho & 0 \\ \rho & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Here, $\sigma_a^2 = \sigma_b^2 = \sigma_c^2 = \sigma^2$.

However, estimation of MNP is very complicated, and there is “identification problem” for covariance matrix.

R library “MNP” is easy to estimate MNP (developed by Kosuke IMAI:今井耕介).
It uses MCMC algorithm. Try it.

Derivation of Nested Logit Model (NL) by GEV:

GEV is "Generalized Extreme Value" models, called "GEV family" proposed by McFadden (1978).

Let $Y_j \equiv exp\left[V_j\right]$ and define "G function" as $G = G\left(Y_1, \dots, Y_J\right)$, $G_j = \frac{\partial G}{\partial Y_j}$.

Train (2003) explains as: "If this function meets certain conditions, then a discrete choice model can be based upon it. In particular, if G satisfies the conditions that are listed below, then

$$P_i = \frac{Y_i G_i}{G}$$

is the choice probability for a discrete choice model that is consistent with utility maximization. Any model that can be derived in this way is a GEV model. This formula therefore defines the family of GEV models."

The properties that the function must exhibit are the following.

1. $G \geq 0$ for all positive values of $Y_j \; \forall \, j$.
2. G is homogeneous of degree one. That is, if each Y_j is raised by some proportion ρ, G rises by proportion ρ also: $G(\rho Y_1, \dots, \rho Y_J) = \rho G(Y_1, \dots, Y_J)$.
3. $G \to \infty$ as $Y_j \to \infty$ for any *j*.
4. The cross partial derivatives of G change signs in a particular way.

$$\frac{\partial^k G}{\partial Y_i^k} \begin{cases} \geq 0 & if \; k \; is \; odd \\ \leq 0 & if \; k \; is \; even \end{cases}$$

Logit model: $G = \sum_{j=1}^{J} Y_j$, $\quad P_i = \frac{Y_i G_i}{G} = \frac{Y_i}{\sum_{j=1}^{J} Y_j} = \frac{e^{V_i}}{\sum_{j=1}^{J} e^{V_j}}$

Nested Logit model: $G = \sum_{l=1}^{K} \left(\sum_{j \in B_l} Y_j^{1/\lambda_l} \right)^{\lambda_l}$

The *J* alternatives are partitioned into *K* nests labeled $B_1, \dots, B_K$.

The *i* alternative belongs to the *k* nest.

$$G_i = \lambda^k \left(\sum_{j \in B_k} Y_j^{1/\lambda_k} \right)^{\lambda_k - 1} \frac{1}{\lambda^k} Y_i^{1/\lambda_k - 1} = Y_i^{1/\lambda_k - 1} \left(\sum_{j \in B_k} Y_j^{1/\lambda_k} \right)^{\lambda_k - 1}$$

$$P_i = \frac{Y_i G_i}{G} = \frac{Y_i Y_i^{1/\lambda_k - 1} \left(\sum_{j \in B_k} Y_j^{1/\lambda_k} \right)^{\lambda_k - 1}}{\sum_{l=1}^{K} \left(\sum_{j \in B_l} Y_j^{1/\lambda_l} \right)^{\lambda_l}} = \frac{e^{V_i/\lambda_k} \left(\sum_{j \in B_k} e^{V_i/\lambda_k} \right)^{\lambda_k - 1}}{\sum_{l=1}^{K} \left(\sum_{j \in B_l} e^{V_j/\lambda_l} \right)^{\lambda_l}}$$

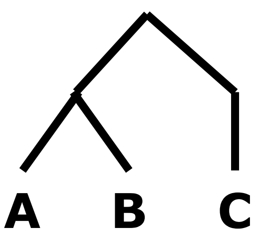
EXERCISE: Please apply & check this equation to three alternatives NL model (ab_c) .

Estimation of Nested Logit (NL):

Derived from Generalized Extreme Value (GEV):

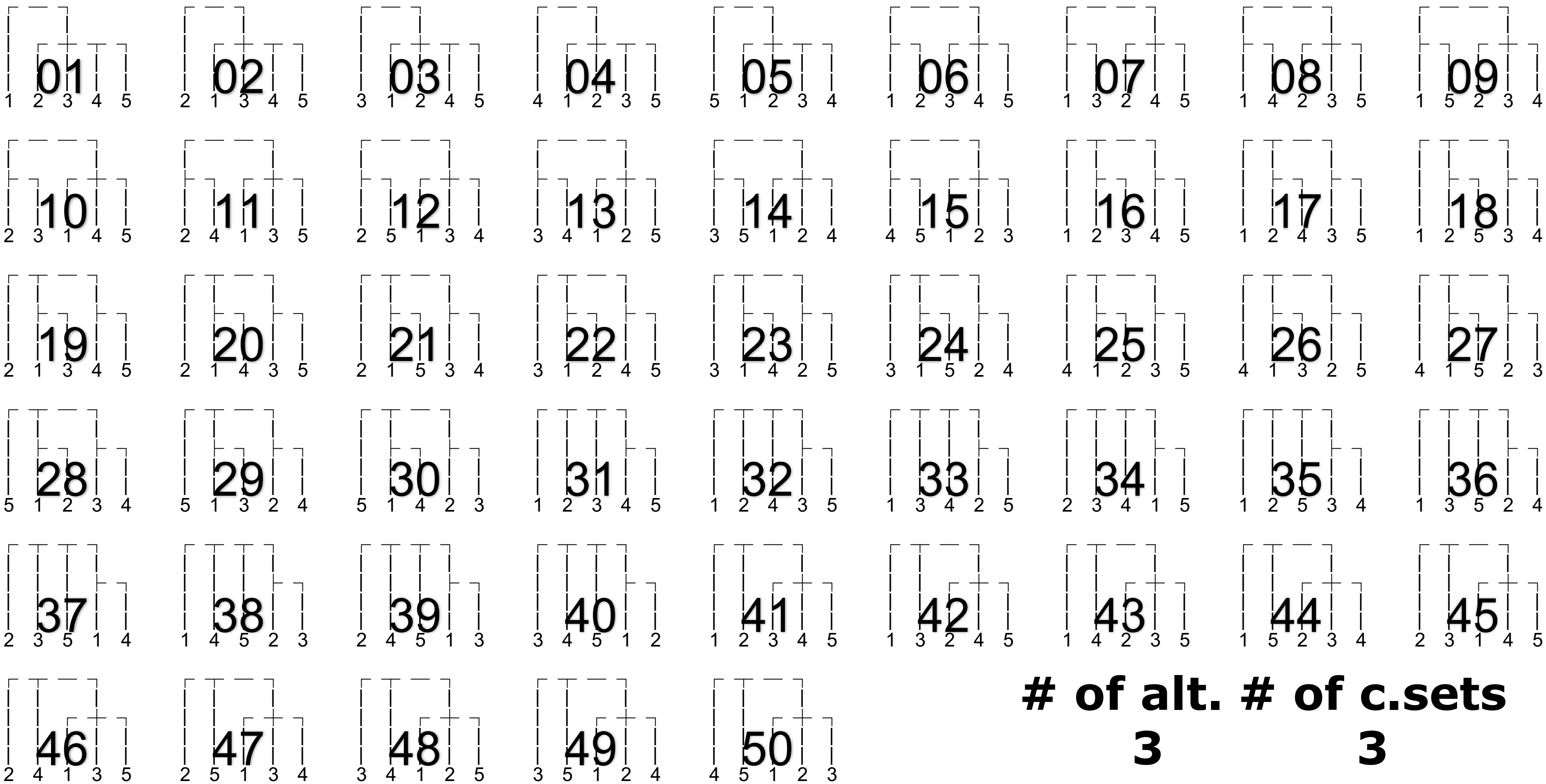
$$P_i = \frac{e^{V_i/\lambda_k}\left(\sum_{j\in B_k} e^{V_j/\lambda_k}\right)^{\lambda_k-1}}{\sum_{l=1}^{K}\left(\sum_{j\in B_l} e^{V_j/\lambda_l}\right)^{\lambda_l}}, \qquad i \in B_k$$

↓ In case of the following NL,



$$P_A = \frac{e^{\frac{V_A}{\lambda}}\left(e^{\frac{V_A}{\lambda}} + e^{\frac{V_B}{\lambda}}\right)^{\lambda-1}}{\left(e^{\frac{V_A}{\lambda}} + e^{\frac{V_B}{\lambda}}\right)^{\lambda} + e^{V_C}} = \frac{e^{\frac{V_A}{\lambda}}}{e^{\frac{V_A}{\lambda}} + e^{\frac{V_B}{\lambda}}} \cdot \frac{e^{\lambda\, ln\left(e^{\frac{V_A}{\lambda}}+e^{\frac{V_B}{\lambda}}\right)}}{e^{\lambda\, ln\left(e^{\frac{V_A}{\lambda}}+e^{\frac{V_B}{\lambda}}\right)} + e^{V_C}}$$

If the number of alternatives is five, two strata NL has 50 choice sets.



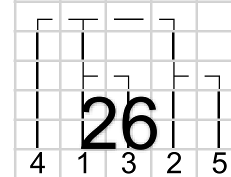
# of alt.	# of c.sets
3	3
4	13
5	50
6	201
7	875
8	3,330

NLcount3.for

NLcount3.for

← FORTRAN program to calibrate this → result. It's very difficult to reach the answer over 8 alternatives case, because this is NP hard problem.

50 NLs estimation code using GEV
This case →



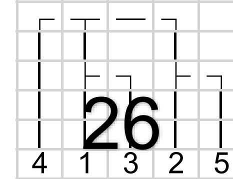
NL.r

NL.r

```
nn <-26; nest.name <- "n_4_13_25"; nl.st[nn] <- nest.name
  n.nest <- 3 ; a.nest <- numeric(n.nest); aloc <- array(0,c(4,4))
  a.nest[1] <- 1; a.nest[2] <- 2; a.nest[3] <- 2
  aloc[1,1] <- 4
  aloc[2,1] <- 1; aloc[2,2] <- 3
  aloc[3,1] <- 2; aloc[3,2] <- 5
  alns <- numeric(5); for(i in 1:n.nest) for(j in 1:a.nest[i])
alns[aloc[i,j]] <- i
```

```
## calculating choice probability
prob <- function(b) {
  u <- matrix(0, nr=ns, nc=n.alt) ## utility function
  p <- matrix(0, nr=ns, nc=n.alt) ## probability
  for(i in 1:n.alt) u[,i] <- exp( (x[,i,] %*% b[1:10])/b[n.var+1] )
  nes <- matrix(0, nr=ns, nc=n.nest)
  for(i in 1:n.nest){
    for(j in 1:a.nest[i]) nes[,i] <- nes[,i] + u[,aloc[i,j]] }
  deno <- numeric(ns)
  for(i in 1:n.nest) deno <- deno + (nes[,i])^b[n.var+1]
  for(i in 1:n.alt){
    p[,i] <- ( u[,i]*(nes[,alns[i]])^(b[n.var+1]-1) ) / deno }
  return(p)
}
```

50 NLs estimation code using mlogit
This case →



mlogit.r

mlogit.r

“d1000.csv” is “wide” format for mlogit

```
library(mlogit)
dt <-read.csv("d1000.csv",header=T)
md <- matrix("",nrow=nrow(dt),ncol=1)
mm <- c("1iba","2tokyo","3hachi",
        "4rail","5seikan")
md <- mm[dt[,1]]; dt[,1] <- md
colnames(dt)[ 2] <- "time.1iba"
colnames(dt)[ 3] <- "time.2tokyo"
colnames(dt)[ 4] <- "time.3hachi"
colnames(dt)[ 5] <- "time.4rail"
colnames(dt)[ 6] <- "time.5seikan"

colnames(dt)[ 7] <- "cost.1iba"
colnames(dt)[ 8] <- "cost.2tokyo"
colnames(dt)[ 9] <- "cost.3hachi"
colnames(dt)[10] <- "cost.4rail"
colnames(dt)[11] <- "cost.5seikan"

colnames(dt)[12] <- "wei1.1iba"
colnames(dt)[13] <- "wei1.2tokyo"
colnames(dt)[14] <- "wei1.3hachi"
colnames(dt)[15] <- "wei1.4rail"
colnames(dt)[16] <- "wei1.5seikan"

colnames(dt)[17] <- "wei2.1iba"
colnames(dt)[18] <- "wei2.2tokyo"
colnames(dt)[19] <- "wei2.3hachi"
colnames(dt)[20] <- "wei2.4rail"
colnames(dt)[21] <- "wei2.5seikan"

colnames(dt)[22] <- "wei3.1iba"
colnames(dt)[23] <- "wei3.2tokyo"
colnames(dt)[24] <- "wei3.3hachi"
colnames(dt)[25] <- "wei3.4rail"
colnames(dt)[26] <- "wei3.5seikan"

colnames(dt)[27] <- "wei4.1iba"
colnames(dt)[28] <- "wei4.2tokyo"
colnames(dt)[29] <- "wei4.3hachi"
colnames(dt)[30] <- "wei4.4rail"
colnames(dt)[31] <- "wei4.5seikan"

m_dt <- mlogit.data(dt,varying=c(2:31),shape="wide",choice="mode")

mnl <- mlogit(mode~time+cost+wei1+wei2+wei3+wei4, m_dt)
summary(mnl)
```

```
b0 <- c(summary(mnl)$CoefTable[,1], 1)

#26
nl_4_13_25 <- mlogit(mode~time+cost+wei1+wei2+wei3+wei4, m_dt, start=b0,
         nests = list( n4 = c('4rail'),n13 = c('1iba','3hachi'),
                       n25 = c('2tokyo','5seikan')), un.nest.el=T)
summary(nl_4_13_25)
para[,26] <- summary(nl_4_13_25)$CoefTable[,1]
tval[,26] <- summary(nl_4_13_25)$CoefTable[,3]
llk[  26] <- summary(nl_4_13_25)$logLik
nl.st[26] <- "4_13_25"
```

Same logsum parameter among nests

```
Coefficients :
                      Estimate Std. Error t-value  Pr(>|t|)
2tokyo:(intercept)  -0.3272063  0.4268821 -0.7665 0.4433772
3hachi:(intercept)   1.4254311  0.4062168  3.5090 0.0004497 ***
4rail:(intercept)   -4.1752487  0.4322056 -9.6603 < 2.2e-16 ***
5seikan:(intercept) -0.6970876  0.3706330 -1.8808 0.0599987 .
time                -0.0142601  0.0116942 -1.2194 0.2226889
cost                -0.0125440  0.0024346 -5.1523 2.573e-07 ***
wei1                 0.0522651  0.1223524  0.4272 0.6692566
wei2                -0.0345885  0.0638537 -0.5417 0.5880357
wei3                -0.7043788  0.1077934 -6.5345 6.381e-11 ***
wei4                 0.5267615  0.1112178  4.7363 2.176e-06 ***
iv                   0.5475420  0.1255754  4.3603 1.299e-05 ***

Log-Likelihood: -1107.6
McFadden R^2:  0.1851
```

NL estimation code using Biogeme

Biogeme → BIerlaire's Optimization routines for GEv Model Estimation

MacOS, linux, Windows version are prepared

http://biogeme.epfl.ch/

Biogeme requires its own data format → basically, tab separated + ended with a space

Microsoft Excel CSV ファイル

d1000bio.csv

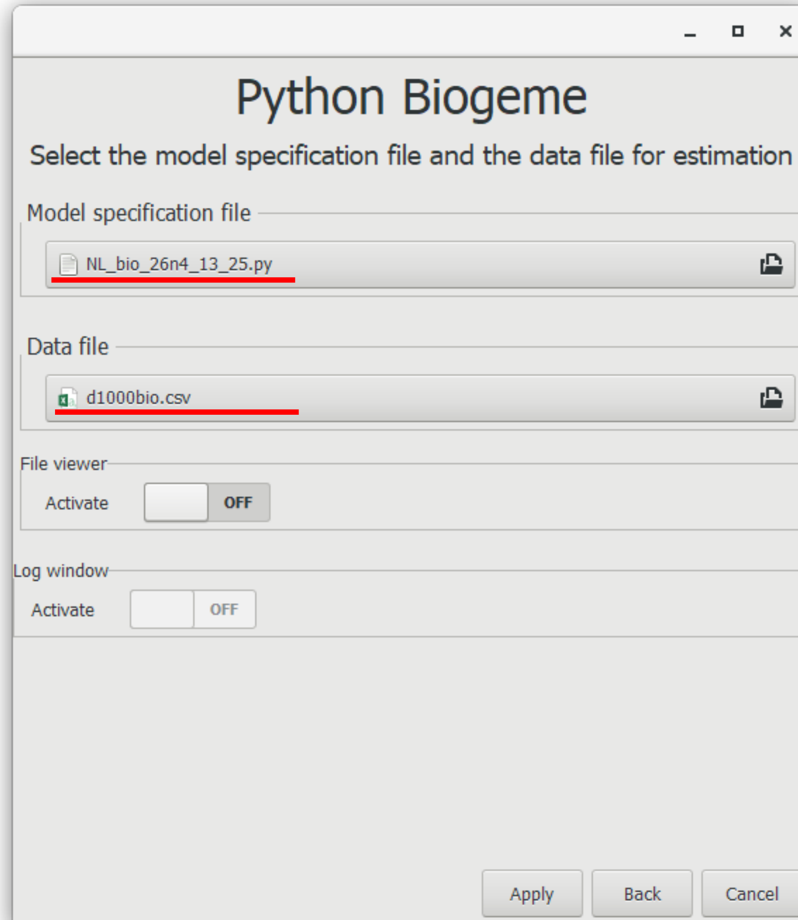
NL_bio_26n4_13_25.py

NL_bio_26n4_13_25.py

```
from biogeme import *
from headers import *
from loglikelihood import *
from statistics import *
from nested import *

ASC_1ibaraki  = Beta('ASC_1ibaraki', 0      , -100, 100, 1,'1ibaraki cte.')
ASC_2tokyo    = Beta('ASC_2tokyo'  ,-0.02391, -100, 100, 0,'2tokyo cte.')
ASC_3hachi    = Beta('ASC_3hachi'  , 2.909  , -100, 100, 0,'3hachi cte.')
ASC_4rail     = Beta('ASC_4rail'   ,-3.905  , -100, 100, 0,'4rail cte.')
ASC_5seikan   = Beta('ASC_5seikan' ,-0.2718 , -100, 100, 0,'5seikan cte.')
B_TIME        = Beta('B_TIME'      ,-0.03043, -100, 100, 0,'Travel time')
B_COST        = Beta('B_COST'      ,-0.01573, -100, 100, 0,'Travel cost')
D_1weight     = Beta('D_1weight'   , 0.4198 , -100, 100, 0,'Weight 1')
D_2weight     = Beta('D_2weight'   , 0.2402 , -100, 100, 0,'Weight 2')
D_3weight     = Beta('D_3weight'   ,-0.8528 , -100, 100, 0,'Weight 3')
D_4weight     = Beta('D_4weight'   , 0.6752 , -100, 100, 0,'Weight 4')
D_5weight     = Beta('D_5weight'   , 0      , -100, 100, 1,'Weight 5')

V1 = ASC_1ibaraki  + B_TIME * time_1ibaraki  + B_COST * cost_1ibaraki + D_1weight * weight_log10
V2 = ASC_2tokyo    + B_TIME * time_2tokyo    + B_COST * cost_2tokyo   + D_2weight * weight_log10
V3 = ASC_3hachi    + B_TIME * time_3hachi    + B_COST * cost_3hachi   + D_3weight * weight_log10
V4 = ASC_4rail     + B_TIME * time_4rail     + B_COST * cost_4rail    + D_4weight * weight_log10
V5 = ASC_5seikan   + B_TIME * time_5seikan   + B_COST * cost_5seikan  + D_5weight * weight_log10
```

```
V = {1: V1, 2: V2, 3: V3, 4: V4, 5: V5 }
av = {1: 1, 2: 1, 3: 1, 4: 1, 5: 1 }

### DEFINITION OF THE NESTS:

# 1: nests parameter
nst = Beta('nst', 1, -100, 100, 0,)
# 2: list of alternatives
m4    = 1/nst, [4]
m13   = 1/nst, [1, 3]
m25   = 1/nst, [2, 5]

nests = m4, m13, m25

logprob = lognested(V,av,nests,mode)
rowIterator('obsIter')
BIOGEME_OBJECT.ESTIMATE = Sum(logprob,'obsIter')

nullLoglikelihood(av,'obsIter')
choiceSet = [1,2,3,4,5]
cteLoglikelihood(choiceSet,mode,'obsIter')
availabilityStatistics(av,'obsIter')

BIOGEME_OBJECT.PARAMETERS['optimizationAlgorithm'] = "BIO"
BIOGEME_OBJECT.FORMULAS['1ibaraki utility'] = V1
BIOGEME_OBJECT.FORMULAS['2tokyo   utility'] = V2
BIOGEME_OBJECT.FORMULAS['3hachi   utility'] = V3
BIOGEME_OBJECT.FORMULAS['4rail    utility'] = V4
BIOGEME_OBJECT.FORMULAS['5seikan  utility'] = V5
```

Definition of Biogeme logsum parameter, λ, is the inverse of original one, the reason is unknown.



Estimation report

Number of estimated parameters:	11
Sample size:	1000
Excluded observations:	0
Init log likelihood:	-1152.136
Final log likelihood:	-1107.643
Likelihood ratio test for the init. model:	88.986
Rho-square for the init. model:	0.039
Rho-square-bar for the init. model:	0.029
Akaike Information Criterion:	2237.286
Bayesian Information Criterion:	2291.272
Final gradient norm:	+8.375e-003
Diagnostic:	Trust region algorithm with simple bounds (CGT2000):
Iterations:	13
Data processing time:	00:00
Run time:	00:03
Nbr of threads:	2

Estimated parameters

Click on the headers of the columns to sort the table [Credits]

Name	Value	Std err	t-test	p-value		Robust Std err	Robust t-test	p-value	
ASC_2tokyo	-0.327	0.379	-0.86	0.39	*	0.370	-0.88	0.38	*
ASC_3hachi	1.43	0.359	3.97	0.00		0.369	3.86	0.00	
ASC_4rail	-4.18	0.429	-9.72	0.00		0.443	-9.43	0.00	
ASC_5seikan	-0.697	0.381	-1.83	0.07	*	0.436	-1.60	0.11	*
B_COST	-0.0125	0.00249	-5.03	0.00		0.00308	-4.07	0.00	
B_TIME	-0.0143	0.0105	-1.36	0.17	*	0.00956	-1.49	0.14	*
D_1weight	0.0523	0.125	0.42	0.68	*	0.141	0.37	0.71	*
D_2weight	-0.0346	0.0656	-0.53	0.60	*	0.0713	-0.48	0.63	*
D_3weight	-0.704	0.0947	-7.43	0.00		0.0933	-7.55	0.00	
D_4weight	0.527	0.112	4.69	0.00		0.118	4.47	0.00	
nst	0.547	0.120	4.57	0.00		0.138	3.96	0.00	

Estimation results of 50 NLs by mlogit, homemade ("NL") and Biogeme.

But this is not "d1000.csv" data, it's the original data analyzed by a graduation thesis. The sample size is 4,204.

LLK: log likelihood
Logsum parameters
VOT: value of time [yen/h]

mlogit is slightly unstable, Biogeme is the strongest.

Homemade may be helpful for understanding ☺

		LLK			logsum			VOT		
	NL_str	mlogit	NL	biogeme	mlogit	NL	biogeme	mlogit	NL	biogeme
36	24_1_3_5		-3592.59	-3588.91		57.0264	100.0000		4,627	4,834
40	12_3_4_5	-3602.15	-3610.74	-3611.13	2517.3845	106.5390	100.0000	1,729	1,760	1,768
49	3_5_124	-3583.22	-3588.10	-3589.02	224.4387	56.6258	46.2000	4,343	3,462	3,872
23	3_14_25	-3649.38	-3649.41	-3649.38	29.0261	27.4054	29.0000	-234	-232	-234
15	45_123	-3660.77	-3653.81	-3652.77	4.8286	18.3827	28.3000	1,765	1,724	1,732
33	25_1_3_4	-3686.04	-3669.37	-3669.37	1.3295	10.6266	10.6000	1,851	1,930	1,934
42	1_3_245	-3651.47	-3651.47	-3651.47	8.7209	8.7584	8.7200	6,282	6,309	6,313
30	5_14_23	-3623.52	-3623.52	-3623.52	3.5268	3.5265	3.5300	862	862	863
38	23_1_4_5	-3637.12	-3637.12	-3637.12	3.1184	3.1185	3.1200	1,530	1,530	1,533
5	5_1234	-3627.82	-3627.82	-3627.82	2.8298	2.8304	2.8300	2,980	2,979	2,983
16	1_23_45	-3656.52	-3656.52	-3656.52	2.5300	2.5304	2.5300	1,918	1,918	1,919
44	1_5_234	-3637.92	-3637.92	-3637.92	2.3686	2.3686	2.3700	2,539	2,539	2,538
50	4_5_123	-3652.88	-3652.88	-3652.88	2.2877	2.2872	2.2900	1,594	1,594	1,592
37	14_2_3_5	-3683.15	-3683.15	-3683.15	1.7340	1.7330	1.7300	1,961	1,961	1,953
22	3_12_45	-3680.94	-3680.94	-3680.94	1.6422	1.6422	1.6400	2,107	2,107	2,109
10	23_145	-3680.83	-3680.83	-3680.83	1.5626	1.5633	1.5600	1,794	1,794	1,796
9	15_234	-3682.19	-3682.19	-3682.20	1.3980	1.3984	1.4000	2,007	2,007	2,005
28	5_12_34	-3682.70	-3682.70	-3682.71	1.3265	1.3262	1.3300	1,902	1,902	1,903
1	1_2345	-3684.58	-3684.58	-3684.58	1.2754	1.2736	1.2800	1,983	1,983	1,985
29	5_13_24	-3685.69	-3685.69	-3685.69	1.1085	1.1087	1.1100	2,140	2,141	2,136
27	4_15_23	-3686.05	-3686.05	-3686.05	1.0710	1.0710	1.0700	1,870	1,870	1,866
8	14_235	-3686.19	-3686.19	-3686.19	0.9829	0.9832	0.9830	1,950	1,950	1,948
3	3_1245	-3686.19	-3686.19	-3686.19	0.9788	0.9788	0.9780	1,918	1,918	1,923
47	2_5_134	-3683.21	-3683.21	-3683.21	0.7889	0.7888	0.7890	1,853	1,853	1,852
7	13_245	-3684.33	-3684.33	-3684.33	0.7814	0.7816	0.7810	1,559	1,560	1,560
43	1_4_235	-3684.67	-3684.67	-3684.67	0.7801	0.7794	0.7800	2,311	2,313	2,305
35	34_1_2_5	-3683.38	-3683.38	-3683.38	0.7624	0.7624	0.7620	1,851	1,851	1,853
18	1_25_34	-3682.42	-3682.42	-3682.42	0.6610	0.6610	0.6610	1,969	1,969	1,966
12	25_134	-3681.64	-3681.64	-3681.64	0.6584	0.6584	0.6580	1,954	1,954	1,956
26	4_13_25	-3674.09	-3674.09	-3674.09	0.5446	0.5446	0.5450	2,386	2,386	2,391
48	3_4_125	-3678.03	-3678.03	-3678.03	0.5334	0.5334	0.5330	2,633	2,633	2,628
39	13_2_4_5	-3672.07	-3672.07	-3672.07	0.5042	0.5042	0.5040	2,065	2,065	2,062
24	3_15_24	-3675.36	-3675.36	-3675.36	0.4461	0.4461	0.4460	880	880	881
4	4_1235	-3676.35	-3676.35	-3676.35	0.4187	0.4187	0.4190	3,630	3,630	3,627
19	2_13_45	-3652.35	-3652.35	-3652.35	0.3620	0.3620	0.3620	1,140	1,140	1,140
31	45_1_2_3	-3673.81	-3665.37	-3665.37	0.4916	0.2801	0.2800	1,828	936	939
45	2_3_145	-3645.47	-3645.47	-3645.47	0.2503	0.2503	0.2500	1,508	1,508	1,510
34	15_2_3_4	-3634.87	-3622.75	-3622.75	0.3705	0.2063	0.2060	3,004	4,081	4,084
25	4_12_35	-3655.78	-3655.78	-3655.78	0.1978	0.1978	0.1980	6,074	6,074	6,065
32	35_1_2_4	-3610.91	-3610.91	-3610.91	0.1110	0.1110	0.1110	10,373	10,373	10,357
21	2_15_34	-3612.94	-3612.93	-3612.93	0.0486	0.0548	0.0543	1,794	1,795	1,789
13	34_125	-3653.68	-3653.45	-3653.43	0.0456	0.0102	-0.0111	2,110	2,130	2,110
17	1_24_35	-3654.10	-3654.09	-3653.99	0.0101	0.0100	-0.0424	124	129	35
41	1_2_345	-3607.81	-3610.74	-3606.75	0.0158	106.5390	-0.0730	1,322	1,760	1,422
46	2_4_135	-3539.70	-3645.47	-3539.70	-0.2678	0.2503	-0.2680	-3,154	1,508	-3,150
6	12_345	-3640.82	-3638.43	-3638.43	-0.0175	-0.2794	-0.2790	1,775	1,381	1,383
20	2_14_35	-3608.85	-3602.04	-3602.04	0.0222	-0.2932	-0.2930	2,233	2,186	2,188
11	24_135	-3579.68	-3579.68	-3579.68	-0.7081	-0.7081	-0.7080	-638	-638	-636
2	2_1345	-3543.79	-3543.79	-3543.79	-0.8587	-0.8588	-0.8590	1,798	1,798	1,796
14	35_124	-3646.92	-3646.92	-3646.92	-1.5225	-1.5220	-1.5200	1,770	1,770	1,767