รายการอ้างอิง

ภาษาไทย

ณัฐวุธ กองสุทธิ. พฤติกรรมการเลือกเส้นทางและความเต็มใจที่จะจ่ายเงินสำหรับระบบแนะนำเส้นทางของผู้ขับขี่ในกรุงเทพมหานคร. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมโยธา บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย 2542.

สมพงษ์ ศิริโสภณศิลป์. แบบจำลองวิเคราะห์การเลือกใช้รถไฟฟ้าขนส่งมวลชนในกรุงเทพมหานคร. ทุนวิจัยรัชดาสมโภช จุฬาลงกรณ์มหาวิทยาลัย 2541.

ภาษาอังกฤษ

Bierlaire, M., Axhausen, K. and Abay, G. The Acceptance of Modal Innovation. The Case of Swissmetro, Proceeding of 1st Swiss Transport Research Conference, 2001a.

Bierlaire, M. A Theoretical Analysis of The Cross-Nested Logit Model. Report RO-011218, Department of Mathemetics, FUNDP, 2001b.

Vovsha, P. Application of Cross-Nested Logit Model to Mode Choice in Tel Aviv, Israel, Metropolitan Area. Transportation Research Record, 1997.

Wen, C.H. and Koppelman, F.S. The Generalized Nested Logit Model. Transportation Research B 35 (7), 2001.

Yamane, Taro. Statistics : An Introductory Analysis. Aoyama Gakuin University, Tokyo, 1088 – 1089, 1973.

ภาคผนวก

ภาคผนวก ก.

แบบสอบถาม

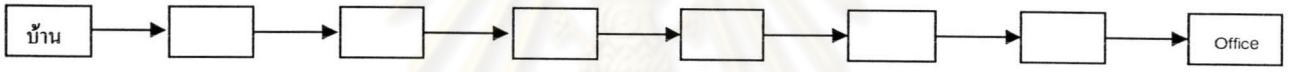
ตัวอย่างที่.....

แบบสอบถามเกี่ยวกับการเดินทางไปทำงาน

- **แบบสอบถามสำหรับทำวิทยานิพนธ์ระดับปริญญาโท <u>คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมโยธา สาขาวิศวกรรมขนส่งและการจราจร จุฬาลงกรณ์มหาวิทยาลัย</u>**
- **ขอกราบขอบพระคุณสำหรับความร่วมมือในการตอบคำถามอย่างละเอียด คำตอบของท่านมีประโยชน์อย่างยิ่งสำหรับผู้ทำวิทยานิพนธ์**

1) ชื่อ.....เบอร์โทรศัพท์ที่ติดต่อได้ (กรุณากรอก).....
2) สถานที่ทำงาน (กรุณากรอกชื่อตึกและถนน).....
3) ที่พักอาศัย (กรุณากรอกอย่างละเอียด).....
.....
4) วันนี้ท่านเดินทางมาทำงานโดยวิธีใด

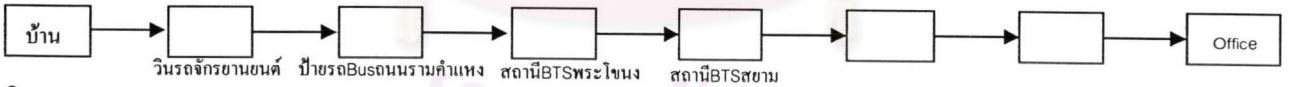
รูปแบบการเดินทาง Walk ....



เวลาการเดินทาง .... นาที
ค่าใช้จ่าย .....0..... .... **บาท**
เวลารอ .... นาที

<u>ตัวอย่าง</u>

รูปแบบการเดินทาง Walk ..... Mor ..... Bus ..... BTS ..... Walk ....



วินรถจักรยานยนต์ ป้ายรถBusถนนรามคำแหง สถานีBTSพระโขนง สถานีBTSสยาม

เวลาการเดินทาง .....1..... .....2..... 45 ..... 20 ..... 3..... ..... นาที
ค่าใช้จ่าย .....0..... .....5..... .....13..... .....18..... .....0..... .... **บาท**
เวลารอ .....0 ..... 10 ..... 3 ..... 0 .... นาที

<u>ข้อมูลเกี่ยวกับผู้เดินทาง</u>

5) เพศ (1) ชาย (2) หญิง
6) อายุ (1) ต่ำกว่า 18 ปี (2) 18 – 25 ปี (3) 26 – 40 ปี
(4) 41 – 60 ปี (5) สูงกว่า 60 ปี
7) รายได้ต่อเดือน (1) ยังไม่มีรายได้ (2) น้อยกว่า 5,000 บาท (3) 5,001 – 10,000 บาท
(4) 10,001 – 20,000 บาท (5) 20,001 – 30,000 บาท (6) 30,001 – 40,000 บาท
(7) 40,001 – 50,000 บาท (8) สูงกว่า 50,000 บาท

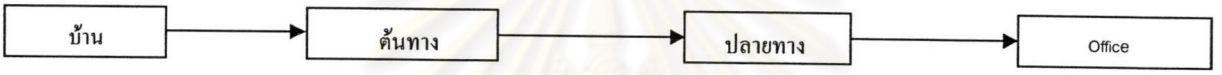
การเดินทางด้วย CAR เป็นการเดินทางหลัก

รูปแบบการเดินทาง Walk,Mor ..... CAR ..... Walk,Mor

เวลาการเดินทาง .... นาที
ค่าใช้จ่าย .... บาท
เวลารอ .... นาที

การเดินทางด้วย BUS เป็นการเดินทางหลัก

รูปแบบการเดินทาง Walk,Mor ..... BUS ..... Walk,Mor



เวลาการเดินทาง .... นาที
ค่าใช้จ่าย .... บาท
เวลารอ .... นาที

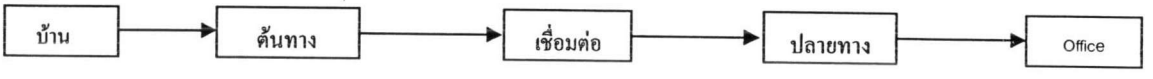
การเดินทางด้วย BTS เป็นการเดินทางหลัก

รูปแบบการเดินทาง Walk,Mor ..... BTS ..... Walk,Mor

เวลาการเดินทาง .... นาที
ค่าใช้จ่าย .... บาท
เวลารอ .... นาที

การเดินทางด้วย BUS-BTS, BTS-BUS เป็นการเดินทางหลัก

รูปแบบการเดินทาง Walk,Mor .... Walk,Mor



เวลาการเดินทาง .... นาที
ค่าใช้จ่าย .... บาท
เวลารอ .... นาที

* ขอกราบขอบพระคุณสำหรับความร่วมมือ *

ภาคผนวก ข.

รายละเอียดของโปรแกรม BIOGEME

1 What's new?

BIOGEME (Version 1.0) includes the following important new features:

1. BIOGEME (Version 1.0) is able to estimate Mixed GEV models. Parameters can arise from a normal, a uniform or a lognormal distribution, and their mean and standard error can be estimated. Normally distributed random parameters can be correlated, and their correlation estimated as well. In this version, log-normally distributed coefficients are permitted taking the exponential of normally distributed parameters.

2. Simulated maximum likelihood estimation is performed with pseudo-random numbers or quasi-random numbers such as Halton sequences.

3. For models with linear-in-parameters utility functions, the attributes are automatically scaled in order to improve the numerical robustness of the model during the estimation. This procedure is entirely transparent to the user.

4. BIOGEME (Version 1.0) gathers in a specific file the results of all model estimations performed in a given directory.

5. BIOGEME (Version 1.0) can now be prematurely interrupted while the estimation algorithm is running, and generate output files corresponding to the best solution found so far.

6. Nonlinear utility functions can be defined by the user. BIOGEME (Version 1.0) will automatically compute the derivatives needed by the optimization algorithm, using automatic differentiation. Due to

the presence of this general functionality, the Box-Cox and Box-Tukey modules, which were available in the previous versions, are now obsolete.

7. The langage to define non trivial expressions has been enhanced. User-defined expressions can now be re-used in other expressions. Moreover, it is possible to define a list of similar expressions using loops.

8. The running time of Multinomial Logit estimation has been significantly decreased.

9. Most functionnalities of BIOGEME (Version 1.0) are now available in the context of panel data.

10. A new optimization algorithm has been implemented.

11. In the presence of a non-identifiable model, BIOGEME (Version 1.0) provides some indications about the problematic parameters.

12. Some bugs have been fixed.

2 Introduction

BIerlaire Optimization toolbox for GEv Model Estimation (BIOGEME) is a freeware designed for the development of research in the context of discrete choice models in general, and of GEV models in particular. All information relative to BIOGEME is maintained at

`roso.epfl.ch/biogeme.`

A users group has been started at

`groups.yahoo.com/group/biogeme`

The documentation is currently limited to this document. If both the document and your "trial-and-errors" do not help solving problems in using the software, make sure you visit the archives of the users group. If you don't find a solution to your problem, post a message to

`groups.yahoo.com/group/biogeme/post`

Also, your contribution is very appreciated. If you have suggestions to improve BIOGEME, post them to the users group as well.

BIOGEME has been developed on Linux, but a Windows version is available. With the distribution of BIOGEME (Version 1.0), there are two additional utilities. BioRoute (Version 1.0) helps preparing the input files for BIOGEME (Version 1.0) in the context of a route choice analysis. It is described in section 15. BioSim (Version 1.0) is designed to perform simulations with a given model. It is described in section 16.

In Section 17, we comment on the examples distributed with BIOGEME (Version 1.0).

We strongly advise to register to Biogeme's users group. If you have any question about BIOGEME (Version 1.0), post them on this group.

3 Invoking BIOGEME (Version 1.0)

BIOGEME (Version 1.0) is invoked in a shell under Linux, in a DOS command window or a Cygwin command window under Windows using the following statement structure

```
biogeme model_name sample_file_1 sample_file2 sample_file3 ...
```

By default, the `sample_file_1` is assumed to be `sample.dat`, and the `model_name` to be `default`. Therefore, typing

```
biogeme model_name
```

is equivalent to typing

```
biogeme model_name sample.dat
```

and typing

```
biogeme
```

is equivalent to typing

```
biogeme default sample.dat
```

Finally, typing

```
biogeme -h
```

generates the following output

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
// BIOGEME Version n.x [Compilation date]
// Author: Michel Bierlaire, EPFL (2001-2004)
// -- Compiled by Michel Bierlaire on Cygwin
// See http://roso.epfl.ch/biogeme
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

"In every non-trivial program there is at least one bug."

Usage: biogeme model_name sampleFile1 sampleFile2 sampleFile3 ...
```

If the name of the model is `mymodel`, say, BIOGEME (Version 1.0) reads the following files:

- a file containing the parameters controlling the behavior of BIOGEME (Version 1.0): `mymodel.par` (section 4)
- a file containing the model specification: `mymodel.mod` (section 5)
- a file containing the data: `sample.dat` (section 6)
- optionnally a file containing the random numbers to use if estimation is based on simulation.

It automatically generates the following output files:

- a file reporting the results of the estimation: `mymodel.rep` (section 9),
- the same file in HTML format,
- a file containing the specification of the estimated model, in the same format as the model specification file: `mymodel.res`
- a file containing some descriptive statistics on the data: `mymodel.sta` (section 8),

and the following files to help understanding possible problems

- a file containing messages produced by BIOGEME (Version 1.0) during the run: `mymodel.log`

- a file containing the values of the parameters which have been actually used by BIOGEME (Version 1.0): `parameters.out`
- a file containing the data stored in BIOGEME (Version 1.0) to represent the model: `model.debug`
- a file containing the specification of the model, as it has actually been understood by BIOGEME (Version 1.0): `_specFile.debug`

These file names may be modified, according to the following rules:

1. If an input file `mymodel.xxx` does not exist, BIOGEME (Version 1.0) attempts to open the file `default.xxx`. If this file does not exist, BIOGEME (Version 1.0) exits with an error. Typically, the parameter file is not model dependent. Therefore, it is common to call it `default.par` to avoid copying it for each different model to be estimated.
2. If an output file `mymodel.xxx` already exists, BIOGEME (Version 1.0) does not overwrite it. Instead, it creates the file `mymodel~1.xxx`. If the file `mymodel~1.xxx` exists, BIOGEME (Version 1.0) creates the file `mymodel~2.xxx`, and so on.

To avoid any ambiguity, BIOGEME (Version 1.0) displays the filenames it has actually used for a specific run, for instance

```
biogeme Input files
===================
Parameters:                          default.par
Model specification:                 mymodel.mod
Sample:                              sample.dat
biogeme Output files
====================
Estimation results:                  mymodel~3.rep
Estimation results (HTML):           mymodel~5.html
Result model spec. file:             mymodel~2.res
Sample statistics:                   mymodel~3.sta
biogeme Debug files
===================
Screen copy:                         mymodel.log
Parameters debug:-                   parameters.out
```

```
Model debug:                    model.debug
Model spec. file debug:         __specFile.debug
```

It is highly recommended to regularly clean the working directory and save the output files in a different place.

4 Parameter file

The parameter file provides the parameters controlling the execution of BIOGEME (Version 1.0). It is not mandatory. If it does not exist, BIOGEME (Version 1.0) uses the default values, and automatically creates a file named `default.par`.

The file is divided into sections, each section containing a list of parameters and their corresponding value.

Section [GEV]

`gevAlgo` It is probably the only parameter in the file that most users will ever want to modify. It selects the optimization algorithm to be used for log-likelihood estimation. As of now, `"BIO"`, `"CFSQP"`, `"SOLVOPT"` and `"DONLP2"` are valid entries. The default is `"BIO"`. More details about these algorithms are available at section 12.

`gevAutomaticScalingOfLinearUtility` If 1, linear utility functions are automatically scaled to avoid numerical problems during the estimation. The scaling is computed in such a way that all attributes have a level of magnitude of about 1.0. Default value: 0.

`gevBinaryDataFile` This is the name of the binary data file where the processed data are stored. Default: `__BiogemeData.bin`.

`gevBufferSize` BIOGEME (Version 1.0) reads the first line of the data files, and stores it in a buffer to analyze it and extract the labels. The size of the buffer is determined by this parameter. The default value is 100'000. Adapt the value if the first line of your data file contains more that 99'999 characters.

`gevCheckDerivatives` If set to 1, the analytical derivatives of the log-likelihood functions and the nonlinear constraints are compared to the finite difference derivatives. This is used basically when a

new model is included and for debugging purposes. Default value: 0.

`gevDataFileDisplayStep` While pre-processing the data file before the estimation, BIOGEME (Version 1.0) reports progress each time it has read a given number of data. This number is specified by the parameter `getDataFileDisplayStep`, and its default value is 500.

`gevDebugDataFirstRow` BIOGEME (Version 1.0) can print what it actually reads from the data file. This parameter is the number of the first row for which his information is displayed. It is recommended to use it when strange results are generated by the package. It helps identifying garbage in the data file, such as strings, for instance. Default: 0.

`gevDebugDataLastRow` BIOGEME (Version 1.0) can print what it actually reads from the data file. This parameter is the number of the last row for which his information is displayed. Default: 0.

`gevDumpDrawsOnFile` If set to 1, BIOGEME (Version 1.0) dumps the draws used for simulated likelihood estimation. The name of the file is displayed at the end of the run. If the model name is `model`, the filename is `model.draws`. Default value: 0.

`gevLogFilePrintLevel` This parameter defines the level of display to be produced in the log file during a run. Valid values are 1 for general messages only, 2 for detailed messages, and 3 for debug messages. Default: 2.

`gevMaxPrimeNumber` The generation of Halton sequences is based on prime numbers. This parameter defines the maximum number of prime numbers that can be used. Most users will never have to change the default value. But if it is too low, an error message is generated:

```
Warning: Error: 23 Halton series must be generated,
but there are only 10 prime numbers available.
Increase the value of gevMaxPrimeNumber in the
parameters file
```

Default value: 1000.

gevMissingValue This parameter is used mainly for debugging purposes. It defines the value given to missing values in the data file. If one of them is used in the computation of the utility functions, an error message is triggered. Default value: 99999.0

gevRandomDistrib There are three valid entries for this parameter: PSEUDO, HESSTRAIN and HALTON. If PSEUDO is selected, maximum simulated likelihood is based on pseudo-random draws. If HALTON is selected, Halton sequences are generated. If HESSTRAIN is selected, a randomized version of systematic sampling is used. Default: PSEUDO

gevScreenPrintLevel This parameter defines the level of display to be produced on the screen during a run. Valid values are 1 for general messages only, 2 for detailed messages, and 3 for debug messages. Default: 1.

gevSeed It defines the seed value for the pseudo-random number generator. Default value: 9021967

gevStopFileName During the optimization process, BIOGEME (Version 1.0) checks for the existence of a file, whose name is defined by this parameter. If the file exists, BIOGEME (Version 1.0) interrupts the iteration and generate output files. This is convenient to prematurely stops iterations without loosing the computations performed thus far. The default value is "STOP"

gevStoreDataOnFile BIOGEME (Version 1.0) uses a database gathering the processed data from the file provided by the users and, if applicable, the draws for the simulated maximum likelihood estimation. If the parameter is 0, the database is stored in memory. If 1, it is stored in the binary file _BiogemeData.bin. It is recommended to use 0, except if the data does not fit in the memory. Indeed, accessing to the file slows down the estimation process. Default: 0.

gevSummaryFile Name of the file summarizing several runs of BIOGEME (Version 1.0). Default value: summary.log

gevSummaryParameters Name of the file containing the name of the parameters whose estimated values must be reported in the summary file. Default value: summary.lis

`gevVarCovarFromBHHH` The computation of the variance-covariance matrix of the estimated parameters using finite difference approximation may take a while for complex models. It is sometimes useful to use the BHHH approximation, which is much faster to compute. If so, set this parameter to 1. It is recommended not to use BHHH in the final model. Default: 0.

`gevTtestThreshold` Set the threshold for the t-test hypothesis tests. If the absolute value of a t-test is less than `gevTtestThreshold`, a symbol $*$ will be appended to the relevant line in the report file (section 9). Default value: 1.96.

`gevWarningSign` When a t-test is not successful, a warning size is displayed in the report file and in the HTML file. This parameter defines the nature of this sign. Default value: *.

Section [BasicTrustRegion]

This section is designed for the BIO optimization algorithm (see section 12).

`BTRTypf` Typical value of the log-likelihood function (see section 12). Default: 1.0.

`BTRTolerance` Value used for the stopping criterion (see section 12). Default: 6.05545e-06.

`BTRCheapHessian` If 1, BHHH (see [Berndt et al., 1974])) is used as an approximation of the second derivatives matrix. Default: 1.

`BTRUsePreconditioner` If 1, the subproblem is preconditioned using a modified Cholesky factorization ([Schnabel and Eskow, 1991]). Default: 1.

Section [cfsqp] This section is designed to define parameters needed by the CFSQP algorithm (section 12).

`cfsqpIprint` Set it to 1 for silent mode, and to 2 for information at each iteration of the optimization algorithm. Default is 1.

`cfsqpMaxIter` Maximum number of iterations. Default is 500

`cfsqpMode` Even if it is a descent algorithm, CFSQP sometimes allows non-monotone iterates, hoping not to be trapped in local minima.

If the function is convex, a descent algorithm is more appropriate. In this case, set the value to 100. See CFSQP manual for more details. Default is 110.

`cfsqpEps` See CFSQP manual. Default is 6.05545e-06. In general, it should not be changed.

`cfsqpEpsEqn` See CFSQP manual. Default is 6.05545e-06. In general, it should not be changed.

`cfsqpUdelta` See CFSQP manual. Default is 0.0. In general, it should not be changed.

Section [solvopt] This section is designed to define parameters needed by the SOLVOPT algorithm (section 12).

`solvoptMaxIter` Maximum number of iterations. Default is 15000.

`solvoptDisplay` Controls the display of the algorithm. See SOLVOPT manual. Default is 1.

`solvoptErrorArgument` See SOLVOPT manual. Default is 1.0e-4. In general, it should not be changed.

`solvoptErrorFunction` See SOLVOPT manual. Default is 1.0e-6. In general, it should not be changed.

Section [donlp2] This section is designed to define parameters needed by the DONLP2 algorithm (section 12).

`donlp2Epsx` See DONLP2 manual. Default is 1.0e-5. In general, it should not be changed.

`donlp2Delmin` See DONLP2 manual. Default is 1.0e-6. In general, it should not be changed.

`donlp2Smallw` See DONLP2 manual. Default is 3.66685e-11. In general, it should not be changed.

`donlp2Epsdif` See DONLP2 manual. Default is 0.0. In general, it should not be changed.

`donlp2NReset` See DONLP2 manual. Default is 9 . In general, it should not be changed.

Name	Value
gevAlgo	BIO
gevAutomaticScalingOfLinearUtility	0
gevBinaryDataFile	__BiogemeData.bin
gevBufferSize	100000
gevCheckDerivatives	0
gevDataFileDisplayStep	500
gevDebugDataFirstRow	0
gevDebugDataLastRow	0
gevDumpDrawsOnFile	0
gevLogFilePrintLevel	2
gevMaxPrimeNumber	1000
gevMissingValue	99999
gevRandomDistrib	PSEUDO
gevScreenPrintLevel	1
gevSeed	9021967
gevStopFileName	STOP
gevStoreDataOnFile	0
gevSummaryFile	summary.log
gevSummaryParameters	summary.lis
gevVarCovarFromBHHH	0
gevTtestThreshold	1.96
gevWarningSign	*

Table 1: Default values of the parameters

It seems that syntax errors in `default.par` cause BIOGEME (Version 1.0) to skip the rest of the file, ignoring all remaining parameters without complaining. This "bug" still has to be fixed. BIOGEME (Version 1.0) writes in the file `parameters.out` the values of the parameters that have been actually used. Make sure you check this file the first times you use BIOGEME (Version 1.0).

5 Model specification file

The file `mymodel.mod` contains the specification of the discrete choice model to be estimated. The sections of this file have to be specified as described below.

`[ModelDescription]` Type here any text that describes the model. It may contain several lines. Each line must be within double-quotes, like this

```
[ModelDescription]
"This is the first line of the model description"
"This is the second line of the model description"
```

`[DataFile]` This section is obsolete in BIOGEME (Version 1.0) and can be safely removed.

`[Choice]` Provide here the formula to compute the identifier of the chosen alternative from the data file. Typically, a "`choice`" entry will be available directly in the file, but any formula can be used to compute it. Assume for example that you have numbered alternatives 100, 200 and 300. But in the data file, they are numbered 1,2 and 3. In this case, you must write

```
[Choice]
100 * choice
```

Any expression described in Section [Expressions] is valid here.

[Weight] Provide here the formula to compute the weight associated to each observation. The weight of an observation will be multiplied to the corresponding term in the log-likelihood function. Ideally, the sum of the weights should be equal to the total number of observations, although it is not required. The file reporting the statistics contains a recommendation to adjust the weights in order to comply with this convention.

[ModelDescription] Provide here any comment describing the model. It will be reported verbatim in the report file. Each line must be between double quotes ".

[Beta] Each line of this section corresponds to a parameter of the utility functions. Five entries must be provided for each parameter:

1. Name: the first character must be a letter (any case) or an underscore (_), followed by a sequence of letters, digits, underscore (_) or dashes (-), and terminated by a white space. Note that case sensitivity is enforced. Therefore varname and Varname would represent two different variables.
2. Default value that will be used as a starting point for the estimation;
3. Lower bound on the valid values[1];
4. Upper bound on the valid values;
5. Status, which is 0 if the parameter must be estimated, or 1 if the parameter has to be maintained at the given value.

Note that this section is independent of the specific model to be estimated, as it captures the deterministic part of the utility function.

```
[Beta]
// Name Value        LowerBound   UpperBound   status
ASC1    -5.22e-02      -1.0          1.0         0
ASC2        0.0        -1.0          1.0         1
```

[1]Bounds specification is mandatory in BIOGEME (Version 1.0). If you do not want bounds, just put large negative values for lower bounds and large positive values for upper bounds. Anyway, if the bound is not active at the solution, it does not play any role, except for safeguarding the algorithm.

```
ASC3    -4.06e-01    -1.0    1.0    0
ASC4          0.0    -1.0    1.0    1
BETA1   -2.06e-02    -1.0    1.0    0
BETA2   -2.19e-02    -1.0    1.0    0
```

[Mu] μ is the homogeneity parameter of the GEV model. Usually, it is constrained to be one. However, BIOGEME (Version 1.0) enables to estimate it if necessary. Four entries are specified here:

1. Default value that will be used as a starting point for the estimation;
2. Lower bound on the valid values;
3. Upper bound on the valid values;
4. Status, which is 0 if the parameter must be estimated, or 1 if the parameter has to be maintained at the given value.

[Utilities] Each row of this section corresponds to an alternative. Four entries are specified:

1. The identifier of the alternative, with a numbering convention consistent with the choice definition;
2. The name of the alternative: the first character must be a letter (any case) or an underscore (_), followed by a sequence of letters, digits, underscore (_) or dashes (-), and terminated by a white space;
3. The availability condition: this must be a direct reference to an entry in the data file, or to an expression defined in the Section [Expressions];
4. The linear-in-parameter utility function is composed of a list of terms, separated by a +. Each term is composed of the name of a parameter and the name of an attribute, separated by a *. The parameter must be listed in Section [Beta], if it is a regular parameter. If it is a random parameter, the syntax is

```
nameParam [ nameParam ]
```

in the case of the normal distribution, or :

```
nameParam { nameParam }
```

to get a random parameter that comes from a uniform distribution. For example, in the case of the normal:

```
BETA [ SIGMA ]
```

Note that the blank after each name parameter is required. Also, parameters BETA and SIGMA have to be listed in Section [Beta]. In the context of an independent random parameter, BETA represents the mean while SIGMA corresponds to the standard deviation. With correlated random parameters, SIGMA technically corresponds to the appropriate term in the Cholesky decomposition matrix that captures the variance-covariance structure among the random parameters. For more details, see the technical section on Cholesky factorization An attribute must be an entry of the data file, or an expression defined in Section [Expressions]. In order to comply with this syntax, the Alternative Specific Constants must appear in a term like ASC * one, where one is defined in the Section [Expressions]. Here is an example:

```
[Utilities]
// Id Name  Avail  linear-in-parameter expression
  1   Alt1   av1   ASC1 * one + BETA1 [SIGMA] * x11 + BETA2 * x12
  2   Alt2   av2   ASC2 * one + BETA1 [SIGMA] * x21 + BETA2 * x22
  3   Alt3   av3   ASC3 * one + BETA1 [SIGMA] * x31 + BETA2 * x32
  4   Alt4   av4   ASC4 * one + BETA1 [SIGMA] * x41 + BETA2 * x42
  5   Alt5   av5   ASC5 * one + BETA1 [SIGMA] * x51 + BETA2 * x52
  6   Alt6   av6   ASC6 * one + BETA1 [SIGMA] * x61 + BETA2 * x62
```

If the utility function does not contain any part which is linear-in-parameters, then the keyword $NONE must be written. For example:

```
[Utilities]
// Id Name  Avail linear-in-parameter expression
  1   Alt1   av1  $NONE
```

[GeneralizedUtilities] This section enables the user to add nonlinear terms to the utility function. For each alternative, the syntax is sim-

ply the identifier of the alternative, followed by the expression. For example, if the utility of alternative 1 is

$$\beta_1 x_{11} + \beta_2 \frac{x_{12}^{\lambda} - 1}{\lambda},$$

the syntax is

```
[Utilities]
1 Alt1 av1 BETA_1 * X11

[GeneralizedUtilities]
1 BETA_2 * (X21 ^ LAMBDA - 1) / LAMBDA
```

Another example where a non-linear part is required is when specifying a log-normal random coefficient. Consult: Example LogNormal.

[ParameterCovariances] BIOGEME (Version 1.0) allows random parameters to be correlated, and can estimate their covariance. By default, the variance-covariance matrix of the random parameters is supposed to be diagonal, and no covariance is estimated. If some covariances must be estimated, each pair of correlated random coefficients must be identified in this section. Each entry of the section should contain:

1. The name of the first random parameter in the given pair. If it appears in the utility function as `BETA [ SIGMA ]`, its name must be typed `BETA_SIGMA`.
2. The name of the second random parameter involved in the pair, using the same naming convention.
3. The default value that will be used as a starting point for the estimation;
4. The lower bound on the valid values;
5. The upper bound on the valid values;
6. The status, which is 0 if the parameter must be estimated, or 1 if the parameter has to be maintained at the given value.

If no covariance is to be estimated, you must either entirely remove the section, or specify `$NONE` as follows:

```
[ParameterCovariances]
$NONE
```

[Draws] Number of draws to be used in Maximum Simulated Likelihood estimation.

[Expressions] In this section are defined all expressions appearing either in the availability conditions or in the utility functions of the alternatives defined in Section [Utilities]. If the expression is readily available from the data file, it can be omitted in the list. We show their use with the help of an example in section 7. As we will discover later in the tutorial, it is good practice to generate new variables from this section especially when one objective is to compute market shares or to evalute effects of policies with the help of BioSim (Version 1.0).

We now summarize the syntax that can be used for generating new variables. Variables which form an expression might be of type float or of type integer. You can use numerical values or the name of a numerical variable. New variables can be created using unary and binary expression operators.

Unary expressions:

1. `y = sqrt(x)` `// y is square root of x.`
2. `y = log(x)` `// y is natural log of x.`
3. `y = exp(x)` `// y is exponential of x.`
4. `y = abs(x)` `// y is absolute value of x.`

binary expression: (Numerical)

1. `y = x + z // y is sum of variables x and z`
2. `y = x - z // y is difference of variables x and z`
3. `y = x * z // y is product of variables x by z`
4. `y = x / z // y is division of variable x by z`
5. `y = x ^ z // y is x to power of z (square would be y = x ^ 2)`
6. `y = x % z // y is x modulo z, i.e. rest of x/z`

binary expression: (Logical)

1. `y = x == z` `// y is 1 if x equals z, 0 otherwise`
2. `y = x != z` `// y is 1 if x not equal to z, 0 otherwise`
3. `y = x || z` `// y is 1 if x != 0 OR z != 0, 0 otherwise`
4. `y = x && z` `// y is 1 if x != 0 AND z != 0, 0 otherwise`
5. `y = x < z` `// y is 1 if x < z (note: also > )`
6. `y = x <= z` `// y is 1 if x <= z (note: also >= )`
7. `y = max(x,z)` `// y is max of x and z (note: also min)`

Note that an expression is considered to be TRUE if it is non zero, and FALSE if it is zero. For a full description of these expressions and alternative syntaxes, please look at the files `patSpecParser.y` and `patSpecScanner.l` in the BIOGEME distribution.

Loops can be defined if several expressions have almost the same syntax. The idea is to replace all occurences of a string, say xx, by numbers. The numbers are generated within a loop, defined by 3 numbers: the start of the loop (a), the end of the loop (b) and the step (c) with the following syntax:

```
{xx a b c}
```

The expression

```
{xx 1 5 2} my_expression_xx = other_expression_xx * term_xx_first
```

is equivalent to

```
my_expression_1 = other_expression_1 * term_1_first
my_expression_3 = other_expression_3 * term_3_first
my_expression_5 = other_expression_5 * term_5_first
```

Warning: make sure that the string is awkward enough so that it cannot match any other instance by mistake. For example, the loop

```
{xp 1 5 2} my_expression_xp = other_expression_xp * term_xp_first
```

is equivalent to

```
my_e1ression_1 = other_e1ression_1 * term_1_first
my_e3ression_3 = other_e3ression_3 * term_3_first
my_e5ression_5 = other_e5ression_5 * term_5_first
```

which is probably not the desired effect.

[Group] Provide here the formula to compute the group ID of the observed individual. Typically, a "group" entry will be available directly from the data file, but any formula can be used to compute it. Any expression described in Section [Expressions] is valid here. A different scale parameter will be estimated for the utility of each group.

[Exclude] Define an expression (see Section [Expressions]) which identifies entries of the data file to be excluded. If the result of the expression is not zero, the entry will be discarded.

[Model] Specifies which GEV model is to be used. Valid entries are $MNL for Multinomial Logit model, $NL for single level Nested Logit model, $CNL for Cross-Nested Logit model and $NGEV for Network GEV model. See section 13 for more details.

[PanelData] Specifies the name of the variable (ex: userID) in the dataset identifying the observations belonging to a given individual and to specify the name of the random parameters that are invariant within the observation of a given individual userID. See the example at subsection 17.12.

[Scale] A scale parameter is associated with each group. The utility function of each member of a group is multiplied by the associated scale parameter. A typical application is the joined estimation of revealed and stated preferences. It is therefore possible to estimate a MNL combining both data sources, without playing around with dummy nested structures as proposed by [Bradley and Daly, 1991]. Each row of this section corresponds to a group. Five entries are required per row:

1. Group number: the numbering must be consistent with the group definition;

2. Default value that will be used as a starting point for the estimation (1.0 is a good guess);
3. Lower bound on the valid values;
4. Upper bound on the valid values;
5. Status, which is 0 if the parameter must be estimated, or 1 if the parameter has to be maintained at the given value.

Clearly, one of the groups must have a fixed scale parameter.

`[NLNests]` This section is relevant only if the `$NL` option has been selected in Section [Model]. If the model to estimate is not a Nested Logit model, the section will be simply ignored. Note that multilevel Nested Logit models must be modeled as Network GEV models. Each row of this section corresponds to a nest. Six entries are required per row:

1. Nest name: the first character must be a letter (any case) or an underscore (_), followed by a sequence of letters, digits, underscore (_) or dashes (-), and terminated by a white space;
2. Default value of the nest parameter μ_m that will be used as a starting point for the estimation (1.0 is a good guess);
3. Lower bound on the valid values. It is usually 1.0, if μ is constrained to be 1.0. Do not forget that, for each nest i, the condition $\mu_i \geq \mu$ must be verified to be consistent with discrete choice theory;
4. Upper bound on the valid values;
5. Status, which is 0 if the parameter must be estimated, or 1 if the parameter has to be maintained at the given value.
6. The list of alternatives belonging to the nest, numbered as specified in Section [Utilities]. Make sure that each alternative belongs to exactly one nest, as no automatic verification is implemented in BIOGEME (Version 1.0).

`[CNLNests]` This section is relevant only if the `$CNL` option has been selected in Section [Model]. If the model to estimate is not a Cross-Nested Logit model, the section will be simply ignored. Note that multilevel Cross-Nested Logit models must be modeled as Network GEV models. Each

row of this section corresponds to a nest. Five entries are required per row:

1. Nest name: the first character must be a letter (any case) or an underscore (_), followed by a sequence of letters, digits, underscore (_) or dashes (-), and terminated by a white space;
2. Default value of the nest parameter μ_m that will be used as a starting point for the estimation;
3. Lower bound on the valid values. It is usually 1.0, if μ is constrained to be 1.0. Do not forget that, for each nest i, the condition $\mu_i \geq \mu$ must be verified to be consistent with discrete choice theory;
4. Upper bound on the valid values;
5. Status, which is 0 if the parameter must be estimated, or 1 if the parameter has to be maintained at the given value.

[CNLAlpha] This section is relevant only if the $CNL option has been selected in Section [Model]. If the model to estimate is not a Cross-Nested Logit model, the section will be simply ignored. Each row of this section corresponds to a combination of a nest and an alternative. Six entries are required per row:

1. Alternative name, as defined in Section [Utilities];
2. Nest name: the first character must be a letter (any case) or an underscore (_), followed by a sequence of letters, digits, underscore (_) or dashes (-), and terminated by a white space;
3. Default value of the parameter capturing the level at which an alternative belongs to a nest that will be used as a starting point for the estimation;
4. Lower bound on the valid values (usually 0.0);
5. Upper bound on the valid values (usually 1.0);
6. Status, which is 0 if the parameter must be estimated, or 1 if the parameter has to be maintained at the given value.

[Ratios] It is sometimes useful to read the ratio of two estimated coefficients. The most typical case is the value-of-time, being the ratio of

the time coefficient and the cost coefficient. This feature is only implemented for fixed parameters. Computation of ratio of random parameters is not permitted in this version. Note that it is not straightforward to characterize the distribution of the ratio of two random coefficients. [Ben-Akiva et al., 1993] suggest a simple approach that is directly implementable in BIOGEME to handle ratio of random parameters. Each row in this section enables to specify such ratios to be produced in the output file. Three entries are required:

1. The parameter (from Section [Beta]) being the numerator of the ratio;
2. The parameter (from Section [Beta]) being the denominator of the ratio;
3. The name of the ratio, to appear in the output file: the first character must be a letter (any case) or an underscore (_), followed by a sequence of letters, digits, underscore (_) or dashes (-), and terminated by a white space.

`[ConstraintNestCoef]` Since Version 0.2, it is possible to constrain nests parameters to be equal. This is achieved by adding to this section expressions like

```
NEST_A = NEST_B
```

where `NEST_A` and `NEST_B` are names of nests defined in Section [NLNests], Section [CNLNests] or Section [NetworkGEVNodes]. This section will become obsolete in future releases, as there is now a section for linear consraints on the parameters: (Section [LinearConstraints]).

`[NetworkGEVNodes]` This section is relevant only if the `$NGEV` option has been selected in Section [Model]. If the model to estimate is not a Network GEV model, the section will be simply ignored. Each row of this section corresponds to a node of the Network GEV model. All nodes of the Network GEV model except the root and the alternatives must be listed here, with their associated parameter. Five entries are required per row:

1. Node name: the first character must be a letter (any case) or an underscore (_), followed by a sequence of letters, digits, underscore (_) or dashes (-), and terminated by a white space;
2. Default value of the node parameter μ_j that will be used as a starting point for the estimation;
3. Lower bound on the valid values. It is usually 1.0. Check the condition on the parameters for the model to be consistent with the theory in [Bierlaire, 2002];
4. Upper bound on the valid values;
5. Status, which is 0 if the parameter must be estimated, or 1 if the parameter has to be maintained at the given value.

`[NetworkGEVLinks]` This section is relevant only if the `$NGEV` option has been selected in Section [Model]. If the model to estimate is not a Network GEV model, the section will be simply ignored. Each row of this section corresponds to a link of the Network GEV model, starting from the a-node to the b-node. The root node is denoted by `__ROOT`. All other nodes must be either an alternative or a node listed in the section [NetworkGEVNodes]. Note that an alternative cannot be the a-node of any link, and the root node cannot be the b-node of any link. Six entries are required per row:

1. Name of the a-node: it must be either `__ROOT` or a node listed in the section [NetworkGEVNodes].
2. Name of the b-node: it must be either a node listed in the section [NetworkGEVNodes], or the name of an alternative.
3. Default value of the link parameter that will be used as a starting point for the estimation;
4. Lower bound on the valid values.
5. Upper bound on the valid values;
6. Status, which is 0 if the parameter must be estimated, or 1 if the parameter has to be maintained at the given value.

`[SampleEnum]` This section is ignored by BIOGEME. It is used by BioSim (Version 1.0) and contains the number of simulations to perform in the sample enumeration step (see Section 16).

[LinearConstraints] In this section, the user can define a list of linear constraints, in one of the following syntaxes:

1. Formula = number,
2. Formula $\leq$ number,
3. Formula $\geq$ number.

The syntax is formally defined as follows:

```
oneConstraint : equation patLESSEQUAL numberParam |
                equation patEQUAL numberParam |
                equation patGREATEQUAL numberParam
equation: eqTerm |
          patMINUS eqTerm |
          equation patPLUS eqTerm  |
          equation patMINUS eqTerm
eqTerm: parameter | numberParam patMULT parameter
```

For example, the constraint

$$\sum_i \mathrm{ASC}_i = 0.0$$

is written

```
ASC1 + ASC2 + ASC3 + ASC4 + ASC5 + ASC6 = 0.0
```

and the constraint

$$\mu \leq \mu_j$$

is written

```
MU - MUJ <= 0.0
```

or

```
MUJ - MU >= 0.0
```

`[NonLinearEqualityConstraints]` In this section, the user can define a list of nonlinear equality constraints of the form

$$h(x) = 0.0.$$

The section must contain a list of functions $h(x)$. For example, the constraint

$$\alpha_{a1}^{\mu_a} + \alpha_{b1}^{\mu_b} = 1$$

is written

```
[NonLinearEqualityConstraints]
ALPHA_A1 ^ MU_A  + ALPHA_B1 ^ MU_B - 1.0
```

`[NonLinearInequalityConstraints]` BIOGEME (Version 1.0) is not able to handle nonlinear inequality constraints yet. It should be available in a future version.

6 Data file

BIOGEME (Version 1.0) assumes that each data file contains in its first line a list of labels corresponding to the available data, and that each subsequent line contains the exact same number of numerical data, each row corresponding to an observation. This number must be the same as the number specified in Section [DataFile]. Delimiters can be tabs or spaces. Note that missing values must not be represented by dots. Instead, replace them by obviously meaningless values, defined by `gevMissingValue`. For those who have used it, the convention was the same in the HieLoW package (see [Bierlaire and Vandevyvere, 1995], [Bierlaire, 1995]).

WARNING: if you have created a data file on DOS or Windows, it may cause problems. If you work in a Windows environnement and want to avoid using Emacs, we recommend using TextPad which is very intuitive to Windows users. Then just make sure you save the file in a UNIX format by selecting the UNIX format in the Save As window. The users working under Linux must convert the file with a utility like `dos2unix`, available from

http://www.megaloman.com/~hany/software/hd2u,

or using Emacs. With GNU Emacs 20.7.1, a (DOS) tag appears at the left of the Emacs info bar when the file is edited, indicating that the file needs to be converted. Use the menu `Mule|Set Coding System|Buffer File` or type[2] `C-x RET f`. Emacs asks you to choose a

"Coding system for visited file (default, nil)".

Choose the default by hitting the return key and save the file.

7 Data transformation

In this section, we use a basic example where the variables x11 to x61 and x12 to x62 are available on the dataset. Let us use the syntax available in the Expression section of the input file to create new variables to be used in the model. The following sections of an input file provide an examples of some new variables that can be created using the Expression section. If one wishes to use BioSim (Version 1.0) to produce the predicted probabilities in order to perform post estimation analysis, it is a good idea to create new variables in the [Expressions] section. Assume that the idea is to see how choice probabilities would vary as the value of x21 is increased, for instance. To get the right calculation, because x21 is involved in the new variables, the change would correctly dissiminate.

```
[Beta]
// Name Value  LowerBound UpperBound  status (0-variable, 1-fixed)
ASC1       0      -10000      10000       1
ASC2       0      -10000      10000       0
ASC3       0      -10000      10000       0
ASC4       0      -10000      10000       0
ASC5       0      -10000      10000       0
ASC6       0      -10000      10000       0
BETA1      0      -10000      10000       0
BETA2      0      -10000      10000       0
GAMMA1     0      -10000      10000       0
GAMMA2     0      -10000      10000       0

[Utilities]
// Id Name  Avail  linear-in-parameter expression (beta1*x1 + beta2*x2 + ... )
   1   Alt1   av1   ASC1 * one + BETA1 * x11 + BETA2 * x12  + GAMMA1 * x11sq   + GAMMA2 * dum12
   2   Alt2   av2   ASC2 * one + BETA1 * x21 + BETA2 * x22  + GAMMA1 * x21sq   + GAMMA2 * dum12
   3   Alt3   av3   ASC3 * one + BETA1 * x31 + BETA2 * x32  + GAMMA1 * x31sq
   4   Alt4   av4   ASC4 * one + BETA1 * x41 + BETA2 * x42  + GAMMA1 * x41sq
   5   Alt5   av5   ASC5 * one + BETA1 * x51 + BETA2 * x52  + GAMMA1 * x51sq
   6   Alt6   av6   ASC6 * one + BETA1 * x61 + BETA2 * x62  + GAMMA1 * x61sq

[Expressions]
```

[2] In Emacs terminology, `C-x` means that you must press the `Ctrl` key and the `x` key together.

```
// Define here arithmetic expressions for name that are not directly
// available from the data
one = 1

        // Loop over alternatives 1 to 6 to create the square of x1i
(zzz 1 6 1)  xzzz1sq = xzzz1 ^ 2

        // Create dum12 = 1 if x11 >= 1 or x21 >= 1, 0 otherwise.
dum12 = ( x11 >= 1 ) || ( x21 >= 1 )
```

8 Statistics

The file containing the statistics of the sample is `mymodel.sta` It contains the following information.

1. The sample size and the sum of all weights are reported. If they don't match, BIOGEME (Version 1.0) suggests a factor to modify the weights:

   ```
   --> It is recommended to multiply all weights by 1.45678
   ```

   In that case, you may want to modify the weight definition in the model specification file:

   ```
   [Weight]
   weight *  1.45678
   ```

2. The number of excluded observations, due to the condition defined in Section [Exclude] of the model specification file, is reported.
3. The total number of observations in the file
4. The number of cases, which is the number of alternatives available to each observation minus the number of observations (see [Ben-Akiva and Lerman, 1985], p. 90).
5. For each attribute, the mean, the minimum and the maximum value across the sample are reported.
6. The number of chosen alternatives, both not taking and taking the weight into account.
7. The group membership, both not taking and taking the weight into account.

9 Report file

The report file (`mymodel.rep`) contains the results of the maximum likelihood estimation of the model. First, general information is reported:

- Type of model which has been estimated.
- Sample size.
- `Null log-likelihood` is the log-likelihood of the sample for a Multinomial Logit model where all β parameters are 0. It is computed as

$$\mathcal{L}^0 = \sum_{n \in \text{sample}} \omega_n \ln \frac{1}{\#\mathcal{C}_n} \tag{1}$$

  where $\#\mathcal{C}_n$ is the number of alternatives available to individual n and ω_n is the associated weight.
- `Init log-likelihood` is the log-likelihood of the sample for the model defined in the `.mod` file.
- `Final log-likelihood` is the log-likelihood of the sample for the estimated model.
- `Likelihood ratio test` is

$$-2(\mathcal{L}^0 - \mathcal{L}^*) \tag{2}$$

  where $\mathcal{L}^0$ is the log-likelihood of the sample for a Multinomial Logit model where all β parameters are 0, defined by (1), and $\mathcal{L}^*$ is the log-likelihood of the sample for the estimated model.
- `Rho-square` is

$$\rho^2 = 1 - \frac{\mathcal{L}^*}{\mathcal{L}^0}. \tag{3}$$

- `Final gradient norm` is the gradient of the log-likelihood function computed for the estimated parameters. If no constraint is active at the solution, it should be close to 0. If there are equality constraints, or if some bound constraints or inequality constraints are active at the solution (that is, they are verified with equality), the gradient may not be close to zero.

ประวัติผู้เขียนวิทยานิพนธ์

นายวิชช์ วิสุทธิประภา เกิดเมื่อวันที่ 25 สิงหาคม พ.ศ. 2523 สำเร็จการศึกษาระดับมัธยมศึกษาจากโรงเรียนเตรียมอุดมศึกษาน้อมเกล้า จังหวัดกรุงเทพฯ สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต (วศ.บ.) ภาควิชาวิศวกรรมโยธา จากจุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2544 และ สำเร็จการศึกษาวิศวกรรมศาสตรมหาบัณฑิต (วศ.ม.) สาขาวิชาวิศวกรรมโยธา ภาควิชาวิศวกรรมโยธา จากจุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2548