

Lab III: Introduction to Biogeme

Meritxell Pacheco
Yuki Oyama, Thibaut Richard

Transport and Mobility Laboratory
School of Architecture, Civil and Environmental Engineering
École Polytechnique Fédérale de Lausanne

October 2, 2018



Outline

- 1 Installation of Biogeme
 - Biogeme
 - Lab computers
 - Download Biogeme
- 2 How does Biogeme work?
 - How to invoke Biogeme
 - Data file
 - Model file
- 3 Today's lab
 - Case study (binary logit)
 - Reading and modifying files



Installation of Biogeme



BIOGEME

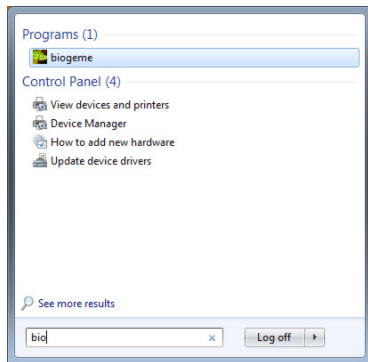


Biogeme

- Created by Michel Bierlaire
- State of the art software for estimating models in the field of discrete choice analysis
- Open source
- All models presented in this course can be estimated with Biogeme
- Webpage: <http://biogeme.epfl.ch>
- **Remark:** We will use the version of Biogeme called **Pythonbiogeme**



Lab computers



- Biogeme is already installed
- Look for it and click on it

Download Biogeme on your own computer

Install

WHAT OS ARE YOU USING?

Biogeme is an open source software. This page explains how to install it on various platforms. In general, it is recommended to install it from sources, so that the software is tailored to your system. However, for users using Windows, a specific installer is also available.

- Install on Mac OS
- Install an executable on Windows
- Install from sources on Windows
- Install from sources

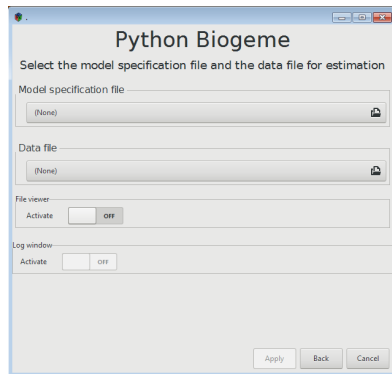
Your OS: MacOS

- Download the program from the **Install** tab
- Follow the instructions related to the OS of your computer

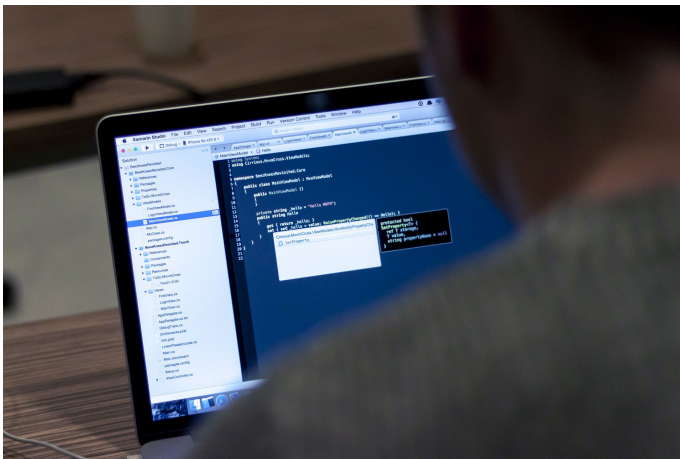


How does the interface look like?

- Windows executable: GUI version
- Mac OS: terminal



How does Biogeme work?

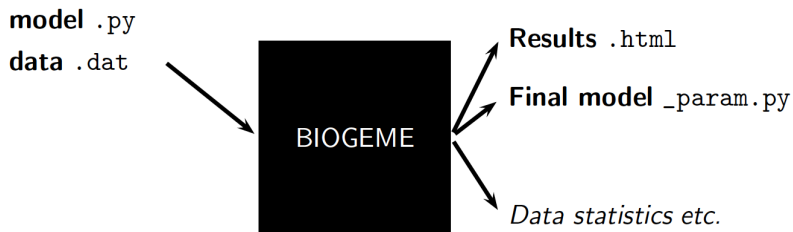


Biogeme files

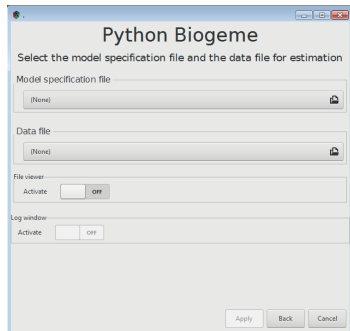
- Biogeme reads:
 - a file containing the model specification `model.py`
 - a file containing the data `data.dat`
- Biogeme automatically generates:
 - A file containing the results of the maximum likelihood estimation:
`model_param.py`
 - The same file in HTML format: `model.html`



How does Biogeme work?



How to invoke Biogeme? (Windows)



- 1 Load the `.py` file in the **Model file** tab
- 2 Load the `.dat` file in the **Data file** tab
- 3 Press **Apply**

How to invoke Biogeme? (Mac OS, Linux)



- Use the terminal
- Access the folder where you have the model and the data files
- Type `pythonbiogeme model data.dat`
 - The name of the model file (without the extension)
 - The name of the data file (with the extension)



Data file (1)

- File extension .dat
- It contains the data, what we call observations
- One observation per row
- First row contains column (variable) names
- Each row must contain a choice indicator
- **Example:** Netherlands transportation mode choice data
 - Choice between rail and car
 - 228 observations



Data file (1)

- File extension .dat
- It contains the data, what we call observations
- One observation per row
- First row contains column (variable) names
- Each row must contain a choice indicator
- **Example:** Netherlands transportation mode choice data
 - Choice between rail and car
 - 228 observations



Data file (2)

netherlands.dat

id	choice	rail_cost	rail_time	car_cost	car_time
1	0	40	2.5	5	1.167
2	0	35	2.016	9	1.517
3	0	24	2.017	11.5	1.966
4	0	7.8	1.75	8.333	2
5	0	28	2.034	5	1.267
...					
219	1	35	2.416	6.4	1.283
220	1	30	2.334	2.083	1.667
221	1	35.7	1.834	16.667	2.017
222	1	47	1.833	72	1.533
223	1	30	1.967	30	1.267

Data file (3)

netherlands.dat

id	choice	rail_cost	rail_time	car_cost	car_time
1	0	40	2.5	5	1.167
2	0	35	2.016	9	1.517
3	0	24	2.017	11.5	1.966
4	0	7.8	1.75	8.333	2
5	0	28	2.034	5	1.267
...					
219	1	35	2.416	6.4	1.283
220	1	30	2.334	2.083	1.667
221	1	35.7	1.834	16.667	2.017
222	1	47	1.833	72	1.533
223	1	30	1.967	30	1.267

Unique identifier of observations

Data file (4)

netherlands.dat

id	choice	rail_cost	rail_time	car_cost	car_time
1	0	40	2.5	5	1.167
2	0	35	2.016	9	1.517
3	0	24	2.017	11.5	1.966
4	0	7.8	1.75	8.333	2
5	0	28	2.034	5	1.267
...					
219	1	35	2.416	6.4	1.283
220	1	30	2.334	2.083	1.667
221	1	35.7	1.834	16.667	2.017
222	1	47	1.833	72	1.533
223	1	30	1.967	30	1.267

Choice indicator, 0: car and 1: train

Model file (1)

- File extension .py
- Must be consistent with the data file
- Contains deterministic utility specifications, model type etc.
- **Example:** Netherlands transportation mode choice data
 - Travel times and travel costs are used as explanatory variables
 - The deterministic utility specifications are

$$V_{\text{car}} = \text{ASC}_{\text{car}} + \beta_{\text{cost}} \text{cost}_{\text{car}} + \beta_{\text{time}} \text{time}_{\text{car}}$$

$$V_{\text{rail}} = \beta_{\text{cost}} \text{cost}_{\text{rail}} + \beta_{\text{time}} \text{time}_{\text{rail}}$$

Model file (2)



- Binary logit: `binary_generic_NL.py` (Netherlands case study)
- Appendix: different parts of the `.py` file and the obtained results

Today's lab



Binary Logit (the Netherlands case study)

- 1 Open the instructions file 03Exercises2018.pdf (under *Instructions*)
- 2 Download the files for this case study (under *Case study*) and copy them in a directory of your choice (e.g., Desktop)
 - binary_generic_NL.py (model file)
 - Binary_Netherlands_2018.pdf (description file)
- 3 The dataset (and its description) can be found in <http://transp-or.epfl.ch/discretechoice/data.html>
- 4 Go through the .py file with the help of the description file
- 5 Run the .py file with Biogeme
- 6 Open the generated .html file and interpret the results
- 7 Develop other model specifications following the instructions file

How can we read and modify files?



- GNU Emacs, TextEdit (Mac), Xcode (Mac), Wordpad (Windows), Notepad++ (Windows)
- **Notepad (Windows) should not be used!**

Appendix

Model file: parameters (binary_generic_NL.py)

```
#Parameters to be estimated
#Arguments:
# 1 Name for report. Typically, the same as the variable
# 2 Starting value
# 3 Lower bound
# 4 Upper bound
# 5 0: estimate the parameter, 1: keep it fixed
ASC_CAR = Beta('ASC_CAR',0,-100,100,0)
ASC_RAIL = Beta('ASC_RAIL',0,-100,100,1)
BETA_COST = Beta('BETA_COST',0,-100,100,0)
BETA_TIME = Beta('BETA_TIME',0,-100,100,0)
```



Model file: expressions (binary_generic_NL.py)

```
#Define here arithm. expressions that are not
#directly available from data
one = DefineVariable('one',1)
rail_time = DefineVariable('rail_time',( rail_ivtt + rail_acc_time )
+ rail_egr_time )
car_time = DefineVariable('car_time', car_ivtt + car_walk_time )
rate_G2E = DefineVariable('rate_G2E', 0.44378022)

car_cost_euro = DefineVariable('car_cost_euro', car_cost * rate_G2E)
rail_cost_euro = DefineVariable('rail_cost_euro', rail_cost * rate_G2E)
```

Model file: utilities (binary_generic_NL.py)

```
#Utility functions
Car = ASC_CAR * one + BETA_COST * car_cost + BETA_TIME * car_time
Rail = ASC_RAIL * one + BETA_COST * rail_cost + BETA_TIME * rail_time

#Which utility functions corresponds to each value of choice
#in the data file
V = {0: Car, 1: Rail}

#Availability conditions for each alternative
av = {0: one, 1: one}

#Excluded observations
BIOGEME_OBJECT.EXCLUDE = sp    != 0
```



Model file: estimation, output (binary_generic_NL.py)

```
#Loglikelihood:
loglik = bioLogLogit(V,av,choice)
#Defines an iterator on the data
rowIterator('obsIter')
#Defines the likelihood function for the estimation
BIOGEME_OBJECT.ESTIMATE = Sum(loglik,'obsIter')
#This is the optimization algorithm used (to compute maximum likelihood)
BIOGEME_OBJECT.PARAMETERS['optimizationAlgorithm'] = "CFSQP"

#Print some statistics:
nullLoglikelihood(av,'obsIter')
choiceSet = [0,1]
cteLoglikelihood(choiceSet,choice,'obsIter')
availabilityStatistics(av,'obsIter')
BIOGEME_OBJECT.FORMULAS['Car utility'] = Car
BIOGEME_OBJECT.FORMULAS['Rail utility'] = Rail
```



Output: .html file (binary_generic_NL.py)

Estimation report

```
Number of estimated parameters: 3
      Sample size: 228
      Excluded observations: 1511
      Init log likelihood: -158.038
      Final log likelihood: -123.133
Likelihood ratio test for the init. model: 69.809
      Rho-square for the init. model: 0.221
      Rho-square-bar for the init. model: 0.202
      Final gradient norm: +4.941e-05
      Diagnostic: Normal termination. Obj: 6.05545e-06 Const: 6.05545e-06
      Iterations: 10
      Data processing time: 00:00
      Run time: 00:00
      Mbr of threads: 1
```

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_CAR	-0.798	0.270	-2.95	0.00	0.275	-2.90	0.00
BETA_COST	-0.0499	0.0103	-4.85	0.00	0.0107	-4.67	0.00
BETA_TIME	-1.33	0.344	-3.86	0.00	0.354	-3.75	0.00

Correlation of coefficients

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

Coefficient1	Coefficient2	Covariance	Correlation	t-test	p-value	Rob. cov.	Rob. corr.	Rob. t-test	p-value
ASC_CAR	BETA_TIME	0.0455	0.491	1.67	0.09	* 0.0464	0.476	1.60	0.11
ASC_CAR	BETA_COST	0.00192	0.693	-2.84	0.00	0.00210	0.713	-2.79	0.01
BETA_COST	BETA_TIME	0.000295	0.0833	3.72	0.00	0.000311	0.0822	3.61	0.00

Smallest singular value: 6.79119

