Computer Lab II
Introduction to Biogeme

Evanthia Kazagli
Anna Fernandez Antolin & Meritxell Pacheco

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

October 4, 2016

Outline

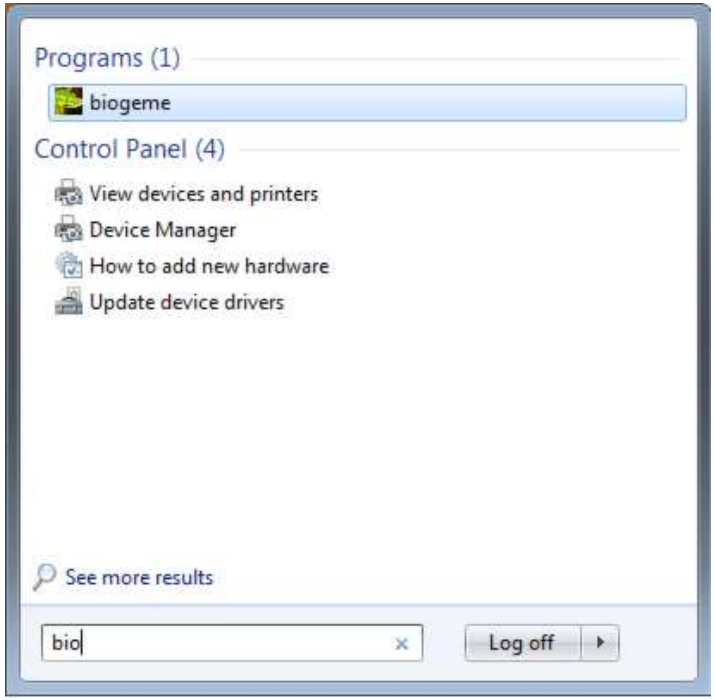
1. Installation of biogeme
2. How does biogeme work?
3. Your work in today's lab

Outline

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

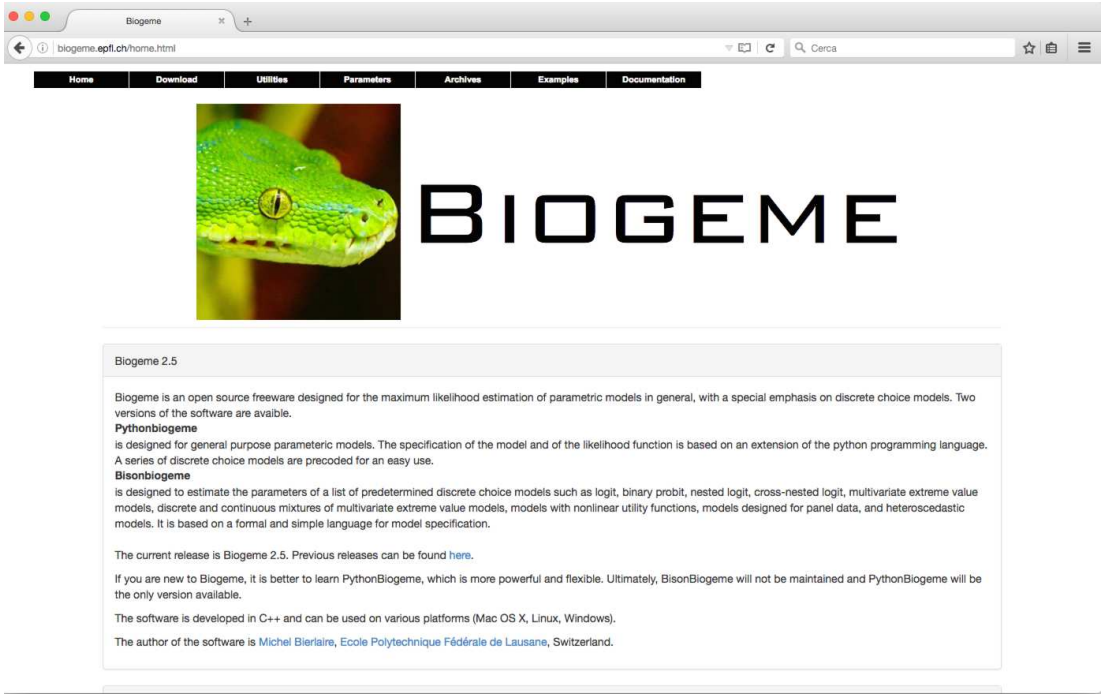
During the labs...



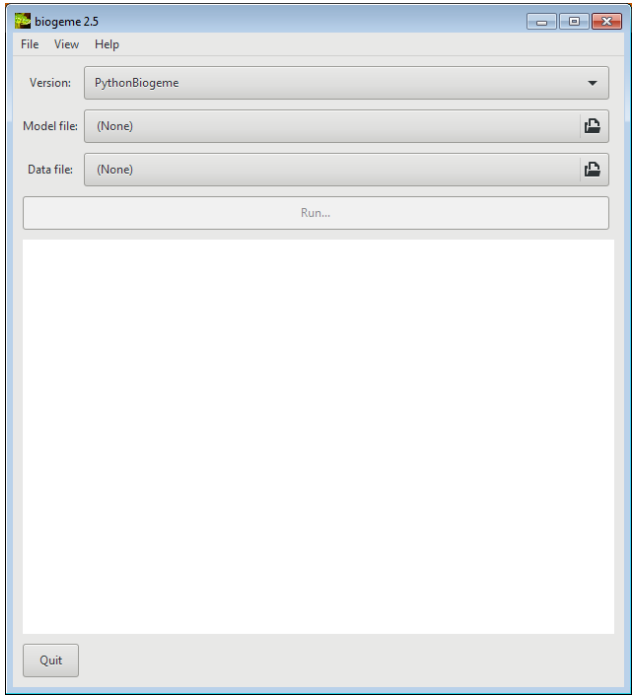
- Biogeme is already installed.
- Look for it and click on it.

Download biogeme on your own computer

- Download the program from the **Download** tab
- Follow the instructions under **Documentation** → **Install**



How does the interface look like?



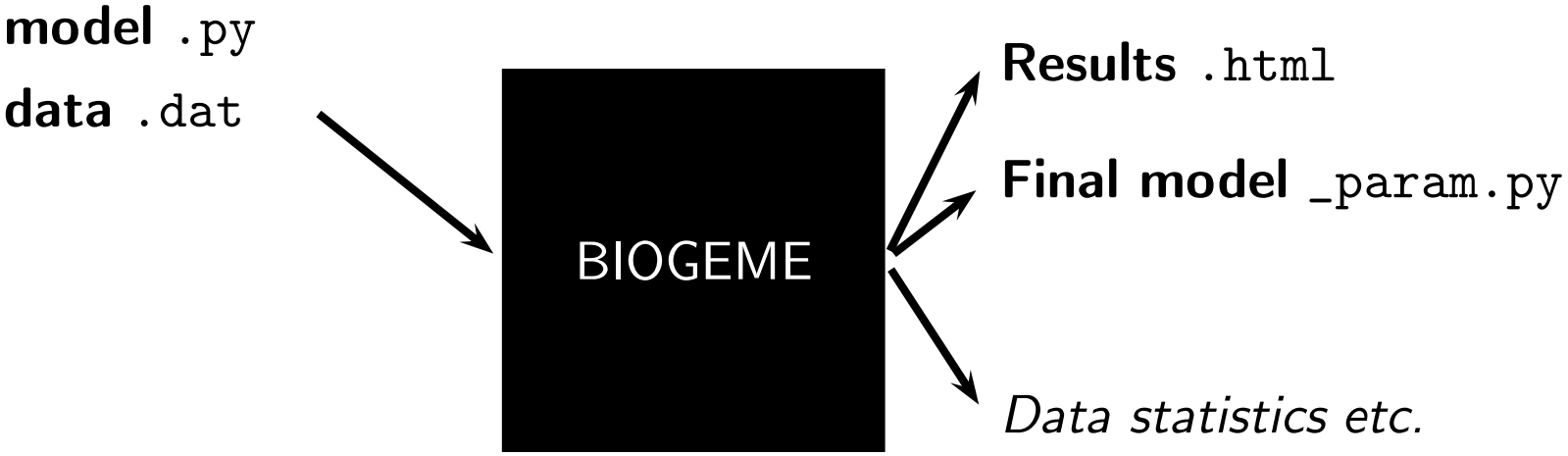
If you work with Mac you can also use the terminal.

Outline

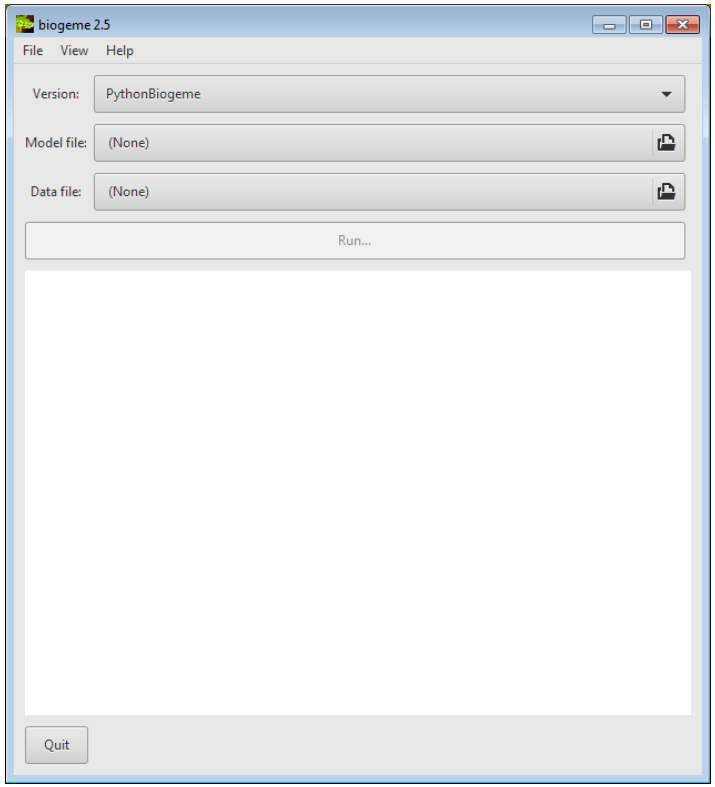
How does BIOGEME work?

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

How does BIOGEME work?



How to invoke Biogeme?



1. Load the .py file in the **Model file** tab
2. Load the .dat file in the **Data file** tab
3. Press **Run..**

Example

- Netherlands mode choice
- Choice between rail and car
- 228 observations
- Travel times and travel costs are used as explanatory variables for the model, and the deterministic utility specifications are

$$
\begin{aligned}
V_{\text{car}} &= \text{ASC}_{\text{car}} + \beta_{\text{cost}}\text{cost}_{\text{car}} + \beta_{\text{time}}\text{time}_{\text{car}} \\
V_{\text{rail}} &= \beta_{\text{cost}}\text{cost}_{\text{rail}} + \beta_{\text{time}}\text{time}_{\text{rail}}.
\end{aligned}
$$

- Model is specified in `model_file.py`

BIOGEME: Data file

- 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 with the Netherlands transportation mode choice data: choice between car and train.

BIOGEME: Data file

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

BIOGEME: Data file

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

BIOGEME: Data file

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

BIOGEME: Model file

- File extension `.py`
- Must be consistent with data file.
- Contains deterministic utility specifications, model type etc.

BIOGEME: Model file

- How can we write the following deterministic utility functions in BIOGEME?

$$
\begin{aligned}
V_{\text{car}} &= \text{ASC}_{\text{car}} + \beta_{\text{time}}\text{time}_{\text{car}} + \beta_{\text{cost}}\text{cost}_{\text{car}} \\
V_{\text{rail}} &= \beta_{\text{time}}\text{time}_{\text{rail}} + \beta_{\text{cost}}\text{cost}_{\text{rail}}
\end{aligned}
$$

BIOGEME: Model file (parameters' section)

```
#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)

# Define here arithm. expressions that are not directly available from data
one  = DefineVariable('one',1)
rail_time  = DefineVariable('rail_time', rail_ivtt   +  rp_rail_ovt  )
car_time  = DefineVariable('car_time', car_ivtt   +  rp_car_ovt  )
```

BIOGEME: Model file (utilities' section)

```
#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 =  (rp   ==  0 )

# The choice model is a logit, with availability conditions. The likelihood is:
prob = bioLogit(__V,__av,choice)
```

BIOGEME: Model file (estimation and output)

```
#And the loglikelihood:
__l = log(prob)
# Defines an itertor on the data
rowIterator('obsIter')
# Defines the likelihood function for the estimation
BIOGEME_OBJECT.ESTIMATE = Sum(__l,'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
```

Estimate your first model

- Download the two files from the course webpage to a directory of your choice (e.g. Desktop).
- Invoke BIOGEME.
- Open the HTML file `model_file.html`.

BIOGEME: Output (Netherlands dataset)

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
                         Nbr 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

Model and Data Files

- How to read and modify model files?
- How to read data files?
  - GNU Emacs, TextEdit (Mac) or Wordpad (Windows)
  - Notepad (Windows) should not be used!

Outline

Binary Logit Case Study

- Available datasets:
  - Airline itinerary choice (Boeing)
  - Mode choice in Netherlands
  - Mode choice in Switzerland (Optima)
- Descriptions available on the course webpage.
- Optima dataset does not contain `.py` files. A specification has to be proposed for the assignment.

How to go through the Case Studies

- Choose a dataset (data descriptions are available on the course webpage).
- Copy the files related to the chosen dataset and case study from the course webpage.
- Go through the `.py` files with the help of the descriptions.
- Run the `.py` files with BIOGEME.
- Interpret the results and compare your interpretation with the one we have proposed.
- Develop other model specifications.

For the assignment

Form groups of ideally four people (groups of three and five will also be accepted).