**CE 6429: Transportation Demand Analysis**

**#Example model of swissmetro**

(source: <https://transp-or.epfl.ch/documents/technicalReports/Bier18.pdf>)

import pandas as pd

import biogeme.database as db

import biogeme.biogeme as bio

import biogeme.models as models

from biogeme.expressions import Beta, DefineVariable

#from biogeme.expressions import Beta, DefineVariable, bioDraws, PanelLikelihoodTrajectory, MonteCarlo, log

#from biogeme.expressions import \*

pandas = pd.read\_table ("swissmetro.dat")

pandas.describe()

database = db.Database ("swissmetro", pandas)

globals().update(database.variables)

exclude = ((PURPOSE != 1)\*(PURPOSE != 3) +\

(CHOICE == 0)) > 0

database.remove(exclude) #(different command)

ASC\_CAR = Beta ('ASC\_CAR', 0, None, None, 0) #from biogeme.expressions import Beta, DefineVariable

ASC\_TRAIN = Beta ('ASC\_TRAIN', 0, None, None, 0)

ASC\_SM = Beta ('ASC\_SM', 0, None, None, 1)

B\_TIME = Beta ('B\_TIME', 0, None, None, 0)

B\_COST = Beta ('B\_COST', 0, None, None, 0)

SM\_COST = SM\_CO \* ( GA == 0 ) #database e ei column dekai na

TRAIN\_COST = TRAIN\_CO \* ( GA == 0 )

CAR\_AV\_SP = DefineVariable ('CAR\_AV\_SP', CAR\_AV \* ( SP !=

0 ), database )

TRAIN\_AV\_SP = DefineVariable ( 'TRAIN\_AV\_SP', TRAIN\_AV \* ( SP

!= 0 ), database )

TRAIN\_TT\_SCALED = DefineVariable ('TRAIN\_TT\_SCALED', TRAIN\_TT / 100.0, database)

TRAIN\_COST\_SCALED = DefineVariable ('TRAIN\_COST\_SCALED', TRAIN\_COST / 100, database )

SM\_TT\_SCALED = DefineVariable ( 'SM\_TT\_SCALED', SM\_TT / 100.0, database)

SM\_COST\_SCALED = DefineVariable ('SM\_COST\_SCALED', SM\_COST / 100, database)

CAR\_TT\_SCALED = DefineVariable ('CAR\_TT\_SCALED', CAR\_TT / 100, database)

CAR\_CO\_SCALED = DefineVariable ('CAR\_CO\_SCALED', CAR\_CO / 100, database)

V1 = ASC\_TRAIN + B\_TIME \* TRAIN\_TT\_SCALED + B\_COST \* TRAIN\_COST\_SCALED

V2 = ASC\_SM + B\_TIME \* SM\_TT\_SCALED + B\_COST \* SM\_COST\_SCALED

V3 = ASC\_CAR + B\_TIME \* CAR\_TT\_SCALED + B\_COST \* CAR\_CO\_SCALED

V = {1: V1, 2: V2, 3: V3}

av = {1: TRAIN\_AV\_SP, 2: SM\_AV, 3: CAR\_AV\_SP }

logprob = models.loglogit (V, av, CHOICE ) #( models.loglogit is for MNL)

biogeme = bio.BIOGEME (database, logprob )

biogeme.modelName = "Multinomial logit"

results = biogeme.estimate()

print (f"HTML file : { results.data.htmlFileName }")