Partial Exam SEN1221 Part 1. December 6 15:45-17:45

An electronics company considers to enter the smartphone market. To better understand the importance of attributes, such as Cost, Size, Memory storage, Camera quality and Operating System (OS) it has hired a high-end consultant to figure out how important these attributes are to consumers of different age groups and genders. Therefore, the consultant has conducted a Stated Choice experiment, in which participants faced 16 hypothetical choice tasks. The screenshot below shows one of the choice tasks. Besides the choice tasks, participants were asked about their age and gender. The data collection has just finished. In total 125, participants have completed the experiment.

	Alternative 1	Alternative 2	Alternative 3
Size [inch]	6.2	5.8	6.4
Storage [GB]	256	128	128
Camera qlt	3	1	4
OS	0	1	0
Cost	800	600	600
Choice	O	O	O

`Coding scheme`

The following coding scheme is used:

OS {0: Android, 1: iOS}

Camera quality {1: mediocre, 2: Good, 3: Very Good, 4: Excellent}

Age {1: Young, 2: Middle age, 3: Old}

Gender {0: Male, 1: Female}

You are tasked to conduct a first analysis of the data.

Run this cell to create your environment locally

In [ ]:
```
# !pip install
```

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import biogeme.biogeme as bio
import biogeme.database as db
from biogeme import models
import biogeme.logging as blog
from biogeme.expressions import (Beta, log, exp, bioDraws, bioMultSum,MonteCarlo, Variable)
import toml
pd.set_option('display.max_columns', 500)
```

In [ ]:
```
# Set the number of draws in the .toml file to 150
# Do not change this code
with open('biogeme.toml', 'r') as file:
    tomldata = toml.load(file)

# Modify the number of draws
tomldata['MonteCarlo']['number_of_draws'] = 150

# Write the modified data back to the .toml file
with open('biogeme.toml', 'w') as file:
    toml.dump(tomldata, file)

# Create a logger to monitor the estimation progress
# if logger does not exist create it, else use it
try:
    logger
except NameError:
    logger = blog.get_screen_logger(level=blog.INFO)
```

Explore the data

In [ ]:
```
# Uncomment this line to load the data in a long format
data  = pd.read_csv('data_partial_exam_long.csv', sep='\t' )
```

In [ ]:
```
data.head(3)
```

Out[ ]:

	ID	CHOICE	COST1	SIZE1	STORAGE1	CAM1	OS1	COST2	SIZE2	STORAGE2	CAM2	OS2	COST3	SIZE3	STORAGE3	CAM3	OS3	GEND
1	1	2	800	6.2	256	3	1	400	6.4	256	4	0	400	5.8	128	1	1	
2	1	3	1000	6.2	256	3	0	600	6.4	64	2	0	800	5.8	512	2	0	
3	1	1	400	5.8	128	4	0	800	6.4	512	1	1	600	6.0	256	2	0	

1 Is this a labelled or unlabelled experiment?

A. Labelled

B. Unlabelled **

2 How is age distributed in the sample?

A. Approximately uniformly distributed

B. Approximately normally distributed

C. Young and old people are the most prevalent groups

D. None of the above **

In [ ]:
```
data['AGE'].value_counts()
```

Out[ ]:
```
AGE
2    986
1    613
3    401
Name: count, dtype: int64
```

3 Are all 3 alternatives available to all decision makers in all observations?

A. Yes **

B. No

In [ ]:
```
data[['AVAIL1','AVAIL2','AVAIL3']].sum()
```

Out[ ]:
```
AVAIL1    2000
AVAIL2    2000
AVAIL3    2000
dtype: int64
```

Estimate a linear-additive RUM-MNL model [Model 1]

- Assume utility is linear and additive for the 5 attributes (hence, treat camera quality as a interval variable)
- Do not include the covariates (i.e AGE or GENDER) in your model

4 What is the final log-likelihood of this model?

A. -2197.23

B -974.54 **

C. -975.25

D. -1112.00

E. -972.01

F. None of the above

5 What is the correct interpretation of the rho sq of this model?

A. The rho square tells us that the data makes the model more likely than throwing a dice **

B. The rho square tells us that this model is too simple to adequately explain the choice behaviour in the data

C. The rho square tells us how likely the data are

D. None of the above **

6 The standard error associated with the betas for the OS and SIZE are larger than 0.05. This tells us that:

A. The OS and SIZE are not significant factors explaining cell phone choices in the population

B. There is substantial heterogeneity between people in their taste for the OS and SIZE

C. On average, people like Apple iOS more than Android OS and like larger phones better than smaller phones

D. None of the above **

In [ ]:
```
# We create the biogeme database
biodata = db.Database('smartphone_data', data)
```

In [ ]:
```
# Attributes of alternative 1
COST1     = Variable('COST1')
SIZE1     = Variable('SIZE1')
STOR1     = Variable('STORAGE1')
CAM1      = Variable('CAM1')
OS1       = Variable('OS1')

# Attributes of alternative 2
COST2     = Variable('COST2')
SIZE2     = Variable('SIZE2')
STOR2     = Variable('STORAGE2')
CAM2      = Variable('CAM2')
OS2       = Variable('OS2')

# Attributes of alternative 3
COST3     = Variable('COST3')
SIZE3     = Variable('SIZE3')
```

```
STOR3     = Variable('STORAGE3')
CAM3      = Variable('CAM3')
OS3       = Variable('OS3')

# Socio-economic variables
AGE       = Variable('AGE')

# The choice
CHOICE    = Variable('CHOICE')

# Give a name to the model
model_name = 'Linear-additive RUM-MNL'

# Define the model parameters, using the function "Beta()", in which you must define:
B_cost  = Beta('B_cost', 0, None, None, 0)
B_size  = Beta('B_size', 0, None, None, 0)
B_stor  = Beta('B_stor', 0, None, None, 0)
B_cam   = Beta('B_cam' , 0, None, None, 0)
B_os    = Beta('B_os'  , 0, None, None, 0)


# Define the utility functions
V1 = B_cost * COST1 + B_size * SIZE1 + B_stor * STOR1 + B_cam * CAM1 + B_os * OS1
V2 = B_cost * COST2 + B_size * SIZE2 + B_stor * STOR2 + B_cam * CAM2 + B_os * OS2
V3 = B_cost * COST3 + B_size * SIZE3 + B_stor * STOR3 + B_cam * CAM3 + B_os * OS3

# Create a dictionary to list the utility functions with the numbering of alternatives
V = {1: V1, 2: V2, 3: V3}

# Create a dictionary called av to describe the availability conditions of each alternative, where 1 indicates that the alterna
# This shows that all alternatives were available to all respondents.
av = {1: 1, 2: 1, 3: 1}

# Define the choice model: The function models.logit() computes the MNL choice probabilities of the chosen alternative given th
prob = models.logit(V, av, CHOICE)

# Define the log-likelihood
LL = log(prob)

# Create the Biogeme object containing the object database and the formula for the contribution to the log-likelihood of each r
biogeme = bio.BIOGEME(biodata, LL)

# The following syntax passes the name of the model:
biogeme.modelName = model_name

# Some object settings regaridng whether to save the results and outputs
```

```python
biogeme.generate_pickle = False
biogeme.generate_html = False
biogeme.saveIterations = False

# Syntax to calculate the null log-likelihood. The null-log-likelihood is used to compute the rho-square
biogeme.calculateNullLoglikelihood(av)

# This line starts the estimation and returns the results object.
results_MNL = biogeme.estimate()

# Print the estimation statistics
print(results_MNL.short_summary())
print(results_MNL.getEstimatedParameters())
```

```
File biogeme.toml has been parsed.
Optimization algorithm: hybrid Newton/BFGS with simple bounds [simple_bounds]
** Optimization: Newton with trust region for simple bounds
Iter.           B_cam          B_cost            B_os          B_size          B_stor      Function    Relgrad   Radius       Rho
    0            0.26         -0.0058           -0.52            0.97          0.0027         1e+03         17    1e+02       1.1
++
    1            0.44         -0.0087           0.025             1.6          0.0043      9.8e+02        5.5    1e+03       1.1
++
    2            0.48         -0.0098           0.11              1.9          0.0046      9.7e+02       0.57    1e+04       1.1
++
    3            0.48         -0.0099           0.15              1.9          0.0047      9.7e+02      0.036    1e+05         1
++
    4            0.48           -0.01           0.15              1.9          0.0047      9.7e+02    4.9e-05    1e+06         1
++
    5            0.48           -0.01           0.15              1.9          0.0047      9.7e+02    0.00027    1e+07         1
++
    6            0.48           -0.01           0.15              1.9          0.0047      9.7e+02    4.8e-07    1e+07         1
++
```

```
Results for model Linear-additive RUM-MNL
Nbr of parameters:              5
Sample size:                    2000
Excluded data:                  0
Null log likelihood:            -2197.225
Final log likelihood:           -974.5448
Likelihood ratio test (null):           2445.36
Rho square (null):                      0.556
Rho bar square (null):                  0.554
Akaike Information Criterion:   1959.09
Bayesian Information Criterion: 1987.094

           Value  Rob. Std err  Rob. t-test  Rob. p-value
B_cam   0.484582      0.032970    14.697745      0.000000
B_cost -0.009951      0.000427   -23.277550      0.000000
B_os    0.148049      0.128148     1.155300      0.247968
B_size  1.888196      0.150574    12.539964      0.000000
B_stor  0.004720      0.000259    18.251420      0.000000
```

`Estimate a new MNL model in which you interact the OS with the three age groups [Model 2].`

- Use this model to infer whether there is a difference between age groups YOUNG, MIDDLE, and OLD regarding their tastes for the OS.
- Assume utility is linear and additive for all 5 attributes (hence, treat camera quality as a interval variable)
- Do not include any other covariates in the model than AGE

`7` What is the final log-likelihood of the model

A. -974.55

B -1134.00

C. -950.95

D. -921.19 **

E. None of the above

`8` Is the model with interactions statistically better than the model without interactions?

The Chi square table is supplied here

A. No

B. Yes, at 10% critical level of significance

C. Yes, at 5% critical level of significance

D. Yes, at 1% critical level of significance **

9 Is there a difference in taste for the OS across the three age groups (Young, Middle, Old)?

A. Yes, the estimated betas for the OS are (significantly) different across age groups **

B. No, the estimated betas for the OS are (almost) similar across all of the age groups

C. It is not possible to tell whether the estimated betas are different from each other across groups

In [ ]:
```
# Give a name to the model
model_name = 'Linear-additive RUM-MNL with interaction'

# Define the model parameters, using the function "Beta()", in which you must define:
B_cost  = Beta('B_cost', 0, None, None, 0)
B_size  = Beta('B_size', 0, None, None, 0)
B_stor  = Beta('B_stor', 0, None, None, 0)
B_cam   = Beta('B_cam' , 0, None, None, 0)
B_os_yng    = Beta('B_os_yng'  , 0, None, None, 0)
B_os_mdl    = Beta('B_os_mdl'  , 0, None, None, 0)
B_os_old    = Beta('B_os_old'  , 0, None, None, 0)


# Define the utility functions
V1 = B_cost * COST1 + B_size * SIZE1 + B_stor * STOR1 + B_cam * CAM1 + (B_os_yng * (AGE==1) + B_os_mdl * (AGE==2) + B_os_old *
V2 = B_cost * COST2 + B_size * SIZE2 + B_stor * STOR2 + B_cam * CAM2 + (B_os_yng * (AGE==1) + B_os_mdl * (AGE==2) + B_os_old *
V3 = B_cost * COST3 + B_size * SIZE3 + B_stor * STOR3 + B_cam * CAM3 + (B_os_yng * (AGE==1) + B_os_mdl * (AGE==2) + B_os_old *

# Create a dictionary to list the utility functions with the numbering of alternatives
V = {1: V1, 2: V2, 3: V3}

# Create a dictionary called av to describe the availability conditions of each alternative, where 1 indicates that the alterna
# This shows that all alternatives were available to all respondents.
av = {1: 1, 2: 1, 3: 1}

# Define the choice model: The function models.logit() computes the MNL choice probabilities of the chosen alternative given th
prob = models.logit(V, av, CHOICE)

# Define the log-likelihood
LL = log(prob)

# Create the Biogeme object containing the object database and the formula for the contribution to the log-likelihood of each r
biogeme = bio.BIOGEME(biodata, LL)

# The following syntax passes the name of the model:
biogeme.modelName = model_name

# Some object settings regaridng whether to save the results and outputs
biogeme.generate_pickle = False
```

```python
biogeme.generate_html = False
biogeme.saveIterations = False

# Syntax to calculate the null log-likelihood. The null-log-likelihood is used to compute the rho-square
biogeme.calculateNullLoglikelihood(av)

# This line starts the estimation and returns the results object.
results_MNL = biogeme.estimate()

# Print the estimation statistics
print(results_MNL.short_summary())
print(results_MNL.getEstimatedParameters())
```

```
File biogeme.toml has been parsed.
Optimization algorithm: hybrid Newton/BFGS with simple bounds [simple_bounds]
** Optimization: Newton with trust region for simple bounds
Iter.           B_cam          B_cost        B_os_mdl        B_os_old        B_os_yng          B_size          B_stor     Functi
on    Relgrad   Radius      Rho
    0             0.3         -0.0059           -0.43           -0.61            0.14             1.4          0.0028        1e+
03         22    1e+02      1.1   ++
    1            0.45         -0.0087           -0.36           -0.87            0.86             1.8          0.0042      9.3e+
02        5.5    1e+03      1.2   ++
    2             0.5          -0.01           -0.29              -1             1.2               2          0.0049      9.2e+
02       0.77    1e+04      1.1   ++
    3            0.51          -0.01           -0.27            -1.1             1.3               2           0.005      9.2e+
02      0.026    1e+05         1   ++
    4            0.51          -0.01           -0.28            -1.2             1.2               2           0.005      9.2e+
02     0.0007    1e+06         1   ++
    5            0.51          -0.01           -0.28            -1.2             1.3               2           0.005      9.2e+
02    3.1e-05    1e+07         1   ++
    6            0.51          -0.01           -0.28            -1.2             1.2               2           0.005      9.2e+
02    8.7e-06    1e+08         1   ++
    7            0.51          -0.01           -0.28            -1.2             1.2               2           0.005      9.2e+
02    7.5e-07    1e+08         1   ++
```

```
Results for model Linear-additive RUM-MNL with interaction
Nbr of parameters:              7
Sample size:                    2000
Excluded data:                  0
Null log likelihood:            -2197.225
Final log likelihood:           -921.1881
Likelihood ratio test (null):           2552.073
Rho square (null):                      0.581
Rho bar square (null):                  0.578
Akaike Information Criterion:   1856.376
Bayesian Information Criterion: 1895.583

             Value  Rob. Std err  Rob. t-test  Rob. p-value
B_cam     0.512427      0.034150    15.005101  0.000000e+00
B_cost   -0.010315      0.000459   -22.471973  0.000000e+00
B_os_mdl -0.281538      0.172259    -1.634391  1.021768e-01
B_os_old -1.159194      0.312707    -3.706970  2.097542e-04
B_os_yng  1.249869      0.168201     7.430784  1.079137e-13
B_size    1.990553      0.153257    12.988340  0.000000e+00
B_stor    0.004980      0.000277    17.984562  0.000000e+00
```

Estimate a linear-additive PANEL Mixed Logit model [Model 3].

- Assume utility is linear and additive for all 5 attributes (hence, treat camera quality as a interval variable)
- Assume tastes for OS are normally distributed in the population: $\beta_{os}^{rnd} \sim N(\beta_{os}, \sigma_{os})$.
- For your convenience, we already prepared the data in a wide format (`data_partial_exam_wide.csv`)
- Note that the data set contains `16` choice observations per individual.

In [ ]:
```
# Uncomment this cell below load the data in a wide format
df_wide = pd.read_csv('data_partial_exam_wide.csv',sep='\t')
biodata_wide = db.Database('data_wide', df_wide)
```

In [ ]:
```
df_wide.head(3)
```

Out[ ]:

	CHOICE_0	COST1_0	SIZE1_0	STORAGE1_0	CAM1_0	OS1_0	COST2_0	SIZE2_0	STORAGE2_0	CAM2_0	OS2_0	COST3_0	SIZE3_0	STO
0	3	800	6.2	256	3	0	600	5.8	128	1	1	600	6.4	
1	2	800	6.2	256	3	0	600	5.8	128	1	1	600	6.4	
2	3	800	6.2	256	3	0	600	5.8	128	1	1	600	6.4	

10 What is the final log-likelihood of the Panel ML model?

Note that in the answers "+/-" means plus or minus 1 LL point

A. -967 +/- 1 **

B -2459 +/- 1

C. -970 +/- 1

D. -819 +/- 1

E. None of the above

11 Based on the results of the Panel ML model, what can you say about heterogeneity in tastes for the OS?

A. The fact that β_{os} is not significant tells us that people in the population don't care about the OS

B The fact that β_{os} is not significant while σ_{os} is significant tells us that only some people care about the OS

C. The fact that β_{os} is not significant while σ_{os} is significant tells us that some people prefer iOS while others prefer Android **

D. The fact that β_{os} is not significant while σ_{os} is significant makes that we cannot say much about the heterogeneity of tastes for the OS **

E. None of the above

12 Given the results of the three models that you have estimated so far, what is the 'best' next model to estimate?

A. A Panel Mixed Logit model which accounts for nesting effects. Thereby, we are able to uncover whether alternatives are correlated in terms of unobserved factors.

B. An MNL model in which we try to interact Gender with tastes, e.g. for size and camera quality. **

C. A Panel Mixed Logit model in which we interact AGE and taste for OS. By combining the insights from Models 2 and 3 we can further refine our understanding of the importance of the OS to different age groups **.

D. A Fully Connected MLP. Thereby, we can see how much variance is unexplained by the current models.

In [ ]:
```
# Number of observations per individual
obs_per_ind = 16

# Define the model parameters
B_cost  = Beta('B_cost', 0, None, None, 0)
B_size  = Beta('B_size', 0, None, None, 0)
B_stor  = Beta('B_stor', 0, None, None, 0)
B_cam   = Beta('B_cam' , 0, None, None, 0)
B_os    = Beta('B_os'  , 0, None, None, 0)
sigma_os = Beta('sigma_os', 1, None, None, 0)

# Construct the random taste parameter for beta_tt
B_os_rnd = B_os + sigma_os * bioDraws('B_os_rnd', 'NORMAL_HALTON2')

# Definition of the utility functions
# Note that we use list comprehension to create a list of utility functions for all observations of an individual
```

```
V1 = [B_cost * Variable(f'COST1_{q}') + B_size * Variable(f'SIZE1_{q}') + B_stor * Variable(f'STORAGE1_{q}') + B_cam * Variable
V2 = [B_cost * Variable(f'COST2_{q}') + B_size * Variable(f'SIZE2_{q}') + B_stor * Variable(f'STORAGE2_{q}') + B_cam * Variable
V3 = [B_cost * Variable(f'COST3_{q}') + B_size * Variable(f'SIZE3_{q}') + B_stor * Variable(f'STORAGE3_{q}') + B_cam * Variable

# Create a dictionary to list the utility functions with the numbering of alternatives
# Note that we use list comprehension to create a list of dictionaries
V = [{1: V1[q], 2: V2[q], 3: V3[q]} for q in range(obs_per_ind)]

# Create a dictionary to describe the availability conditions of each alternative
av = {1:1, 2:1, 3:1}

# Give the model a name
model_name = 'ML with normal distributed B_os'

# The conditional probability of the chosen alternative is a logit
condProb = [models.loglogit(V[q], av, Variable(f'CHOICE_{q}')) for q in range(obs_per_ind)]

# Take the product of the conditional probabilities
condprobIndiv = exp(bioMultSum(condProb))   # exp to convert from logP to P again

# The unconditional probability is obtained by simulation
uncondProb = MonteCarlo(condprobIndiv)

# The Log-likelihood is the log of the unconditional probability
LL = log(uncondProb)

# Create the Biogeme estimation object containing the data and the model
biogeme = bio.BIOGEME(biodata_wide , LL)

# Set reporting levels
biogeme.generate_pickle = False
biogeme.generate_html = False
biogeme.saveIterations = False
biogeme.modelName = model_name

# Compute the null loglikelihood for reporting
# Note that we need to compute it manually, as biogeme does not do this for panel data
biogeme.nullLogLike = len(biodata_wide.data)*np.log(1/3)*obs_per_ind

# Estimate the parameters and print the results
results = biogeme.estimate()
print(results.short_summary())

# Get the results in a pandas table
beta_hat = results.getEstimatedParameters()
print(beta_hat)
```

```
File biogeme.toml has been parsed.
Optimization algorithm: hybrid Newton/BFGS with simple bounds [simple_bounds]
** Optimization: Newton with trust region for simple bounds
Iter.           B_cam          B_cost            B_os          B_size          B_stor        sigma_os     Function    Relgrad
Radius      Rho
    0               0               0               0               0               0               1      1.8e+03         97
5   -0.095    -
    1             1.1          -0.013            -4.2               4          0.0094              -4      1.3e+03         36
5     0.25    +
    2             1.1          -0.013            -4.2               4          0.0094              -4      1.3e+03         36
2.5    -0.28    -
    3           0.098         -0.0079            -3.5             1.5          0.0017            -4.2      1.2e+03         28
2.5     0.31    +
    4            0.65         -0.0092           -0.95             2.6          0.0051            -2.8        1e+03         12
2.5     0.76    +
    5            0.41         -0.0087           -0.37               1          0.0041           -0.33      9.9e+02       0.95
2.5     0.59    +
    6            0.41         -0.0087           -0.37               1          0.0041           -0.33      9.9e+02       0.95
1.2    -0.75    -
    7            0.53           -0.01           -0.18             2.3          0.0051            -1.5      9.8e+02        4.7
1.2     0.25    +
    8            0.48         -0.0097           -0.14             1.7          0.0047           -0.22      9.8e+02        2.8
1.2      0.2    +
    9            0.48         -0.0097           -0.14             1.7          0.0047           -0.22      9.8e+02        2.8
0.62    -0.92    -
   10            0.51          -0.011            0.43             1.9          0.0052           -0.85      9.7e+02       0.48
0.62     0.24    +
   11            0.49           -0.01          -0.022             1.9          0.0048           -0.62      9.7e+02       0.57
0.62      0.8    +
   12             0.5           -0.01            0.02             1.9          0.0048           -0.73      9.7e+02      0.031
6.2         1   ++
   13             0.5           -0.01           0.026             1.9          0.0048           -0.72      9.7e+02     0.0014
62         1   ++
   14             0.5           -0.01           0.026             1.9          0.0048           -0.72      9.7e+02    7.8e-05
6.2e+02         1   ++
   15             0.5           -0.01           0.026             1.9          0.0048           -0.72      9.7e+02    2.4e-06
6.2e+02         1   ++
```

```
Results for model ML with normal distributed B_os
Nbr of parameters:              6
Sample size:                    125
Excluded data:                  0
Null log likelihood:            -2197.225
Final log likelihood:           -967.5042
Likelihood ratio test (null):           2459.441
Rho square (null):                      0.56
Rho bar square (null):                  0.557
Akaike Information Criterion:   1947.008
Bayesian Information Criterion: 1963.978

             Value  Rob. Std err  Rob. t-test  Rob. p-value
B_cam     0.500526      0.034109    14.674381  0.000000e+00
B_cost   -0.010107      0.000462   -21.886907  0.000000e+00
B_os      0.026505      0.152708     0.173564  8.622083e-01
B_size    1.937951      0.156023    12.420930  0.000000e+00
B_stor    0.004829      0.000266    18.140071  0.000000e+00
sigma_os -0.723378      0.134277    -5.387217  7.155710e-08
```