RUMdesignSimulator

RUMdesignSimulator

The package RUMdesignSimulator proposes convenient tools for generating synthetic data for decision theory. Firstly, Alternatives, Decision Makers and Preference Coefficients are easily generated. Then, experimental designs are generated in format long or wide. In addition, the effect of each variable can be visualized in a 3D graph.

Installation:

devtools::install_github("AntoineDubois/RUMdesignSimulator")

Tutorial

This is an R Markdown present the package RUMdesignSimulator. This package generates experimental designs from real data or from probabilistic distributions. First of all, we need to install and load the package. One should install the package devtools if necessary.

```
#devtools::install_github("AntoineDubois/RUMdesignSimulator")
library(RUMdesignSimulator)
```

Now, we can define the setup of the experiment. Thus, we define the name of the alternatives as well as their attributes. In addition, we define the number of decision makers as well as their characteristics.

```
DM_att_names <- list("X1", "X2", "X3") # the list of the decision makers' characteristics
AT_names <- list("good1", "good2", "good3", "good4") # the list of the alternatives' names
AT_att_names <- list("Z1", "Z2", "Z3") # the list of the alternatives' attributes
groups <- c(10, 20) # the groups of decision makers
```

Then, we initialize the instance of the class Experiment. Furthermore, Experiment is a *reference class*. This type of class is the most flexible embeded in R.

```
FD <- Experiment(DM_att_names=DM_att_names, AT_att_names=AT_att_names, AT_names=AT_names,
                 groups=groups, no_choice=TRUE) # creation of an instance of the call Experiment
```

Since the instance FD is an instance of the class Experiment, we can use methods to generate decision makers characteristics according to distributions or from data. To know which laws are implemented in this package, we use the function **information**.

```
information()
```

```
## The available distributions and the name of their respective parameters
```

```
##         Distributions          Parameters
## 1              normal              mu, sd
## 2             student location, scale, df
## 3    discrete_uniform                a, b
## 4  continuous_uniform                a, b
## 5                beta shape1, shape2, ncp
## 6              gumbel     location, scale
## 7                chi2             df, ncp
## 8             poisson              lambda
## 9                 exp              lambda
## 10          empirical                data
## 11               help                None
```

We have chosen the distributions underneath for generating decision makers characteristics.

```r
 # the characteristics of X1 are drawn from a data set
FD$gen_DM_attributes("empirical", data = data.frame(X1=c(0.5, 0, 12, 6, 7.3)), which = "X1")
```

```
## Warning: There remain some NA values in the decision makers' attributes matrix
```

```r
# the characteristics X2 and X3 follow a standardized normal distribution within the group 1
FD$gen_DM_attributes("normal", which=c("X2", "X3"), group=1)
```

```
## Warning: There remain some NA values in the decision makers' attributes matrix
```

```r
# the characteristics X2 and X3 follow a normal distribution with mean 1 and 2 standard deviation withi
FD$gen_DM_attributes("normal", mu=1, sd=2, which=c("X2","X3"), group=2)

FD$X
```

```
##      X1          X2           X3
## 1   0.0  0.37491967  1.402212008
## 2   7.3 -0.02370712 -1.525556071
## 3   6.0 -0.90596375 -0.085875669
## 4   7.3 -1.60760254  1.196754448
## 5  12.0 -0.75502427  0.005916605
## 6  12.0  0.15902388  0.531619934
## 7  12.0 -0.44046232  0.329469585
## 8   0.5  1.84621300 -0.572216985
## 9   6.0  0.38699811 -0.737464079
## 10 12.0  0.91003142  1.793027387
## 11  6.0  0.61756626 -0.328758940
## 12 12.0  0.01361925 -0.306938535
## 13  6.0  1.46411837  2.044019544
## 14 12.0  3.08442833  1.180074941
## 15  0.0 -0.19968193  1.183372943
## 16  6.0 -0.15259895 -1.474325284
## 17  6.0  1.87169211  1.939638613
## 18  0.5  1.44656034  0.706857737
## 19  6.0  0.31507136  3.629598944
## 20  7.3  2.61585173  1.769405931
## 21  7.3  0.45778099  0.678654653
```

```
## 22 12.0 -0.85527957  2.210397776
## 23 12.0  1.09144932 -2.485149094
## 24  0.0  0.65765812  0.956857387
## 25  0.5  1.57608016  0.657172173
## 26 12.0  1.75666394  1.799784572
## 27  6.0 -0.54037639  1.442492088
## 28  7.3  1.77202241  0.838474616
## 29  0.0  1.87410132 -0.514290747
## 30 12.0  0.18198818  0.657430584
```

In addition, we can observe cross efffects between the decision makers' characteristics.

```
FD$gen_DM_attributes(observation=~X1+X2+X3+I(X1*X2))
FD$X
```

```
##      X1          X2           X3  I(X1 * X2)
## 1   0.0  0.37491967  1.402212008   0.0000000
## 2   7.3 -0.02370712 -1.525556071  -0.1730620
## 3   6.0 -0.90596375 -0.085875669  -5.4357825
## 4   7.3 -1.60760254  1.196754448 -11.7354986
## 5  12.0 -0.75502427  0.005916605  -9.0602913
## 6  12.0  0.15902388  0.531619934   1.9082865
## 7  12.0 -0.44046232  0.329469585  -5.2855478
## 8   0.5  1.84621300 -0.572216985   0.9231065
## 9   6.0  0.38699811 -0.737464079   2.3219887
## 10 12.0  0.91003142  1.793027387  10.9203771
## 11  6.0  0.61756626 -0.328758940   3.7053976
## 12 12.0  0.01361925 -0.306938535   0.1634310
## 13  6.0  1.46411837  2.044019544   8.7847102
## 14 12.0  3.08442833  1.180074941  37.0131400
## 15  0.0 -0.19968193  1.183372943   0.0000000
## 16  6.0 -0.15259895 -1.474325284  -0.9155937
## 17  6.0  1.87169211  1.939638613  11.2301527
## 18  0.5  1.44656034  0.706857737   0.7232802
## 19  6.0  0.31507136  3.629598944   1.8904282
## 20  7.3  2.61585173  1.769405931  19.0957176
## 21  7.3  0.45778099  0.678654653   3.3418012
## 22 12.0 -0.85527957  2.210397776 -10.2633548
## 23 12.0  1.09144932 -2.485149094  13.0973918
## 24  0.0  0.65765812  0.956857387   0.0000000
## 25  0.5  1.57608016  0.657172173   0.7880401
## 26 12.0  1.75666394  1.799784572  21.0799673
## 27  6.0 -0.54037639  1.442492088  -3.2422584
## 28  7.3  1.77202241  0.838474616  12.9357636
## 29  0.0  1.87410132 -0.514290747   0.0000000
## 30 12.0  0.18198818  0.657430584   2.1838581
```

Similarly, we generate alternatives' attributes

```
# generation of a random covariance matrix of size 3
sigma <- clusterGeneration::genPositiveDefMat(3)$sigma

# all the attributes are generated by a multivariate normal distribution of mean (-1, 2, 0) and covaria
```

```r
FD$gen_AT_attributes(mu=c(-1,2,0), sd=sigma)

# observation of complex effects between the alternatives' attributes
FD$gen_AT_attributes(observation=~Z1+Z2+Z3+I(Z1^2))

FD$Z
```

```
##                   Z1       Z2          Z3    I(Z1^2)
## no_choice  0.0000000 0.000000  0.00000000 0.00000000
## good1     -0.2033049 2.479220 -0.08550907 0.04133289
## good2     -1.9499097 1.743730 -0.61155024 3.80214791
## good3     -2.2376525 1.619594 -2.04124859 5.00708872
## good4     -0.1964412 2.015248 -0.07676257 0.03858915
```

and decision makers' preferences

```r
#Generation of beta whose components law's are different:

# generation of the variables from 1 to 4 of the alternatives within the group 1
FD$gen_preference_coefficients("student", heterogeneity=TRUE, location=-2,  scale=1, df=4, which=c(1:4)
```

```
## Warning: There remain some NA values in the decision makers' attributes matrix
```

```r
# generation of the variables from 1 to 4 of the alternatives within the group 2
FD$gen_preference_coefficients("student", heterogeneity=FALSE, location=2,  scale=1, df=4, which=c(1:4)
```

```
## Warning: There remain some NA values in the decision makers' attributes matrix
```

```r
# generation of the fifth variable within every group
FD$gen_preference_coefficients("normal", heterogeneity=FALSE, mu=0, sd=2, which=5)
```

```
## Warning: There remain some NA values in the decision makers' attributes matrix
```

```r
# rectification, the variable Z2 follows a discrete uniform distribution
FD$gen_preference_coefficients("discrete_uniform", heterogeneity=TRUE, a=1, b=5, which="Z2")
```

```
## Warning: There remain some NA values in the decision makers' attributes matrix
```

```r
# generation of the variable Z3 and I(Z1^2) according to the default distribution: the standardized nor
FD$gen_preference_coefficients(heterogeneity=TRUE, which=c("Z3", "I(Z1^2)"))

FD$beta
```

```
##           X1        X2         X3  I(X1 * X2)        Z1 Z2          Z3     I(Z1^2)
## 1  -4.6028272 -2.990994 -1.0295190 -1.32566958 -1.974666  4 -1.27086896  0.03264062
## 2  -0.6126987 -1.405607 -4.9254097 -1.60498706 -1.974666  4  0.46007091  0.51755751
## 3  -2.0114217 -2.700239 -2.3957645 -0.86312681 -1.974666  4  2.33733594 -0.18249097
## 4  -1.9390055 -2.230530 -3.7762441 -0.09674584 -1.974666  3 -0.15655132 -1.01669912
## 5  -1.1519141 -2.896638 -0.7476390 -1.97689678 -1.974666  3 -0.76879841 -0.81543360
```

```
## 6  -2.5297844  1.070922 -3.0834410 -1.76293185 -1.974666  1 -0.72544430 -0.24173874
## 7  -2.2326510 -3.362418 -4.3572177 -2.00835688 -1.974666  5 -0.58237497  1.22543297
## 8  -1.1946834 -2.735738 -0.8149382 -0.49639166 -1.974666  3 -0.43919387  1.29704553
## 9  -3.3975202 -2.904391 -3.5985592 -2.50128414 -1.974666  5 -0.21216834  0.03515240
## 10 -0.8366639 -2.817267 -0.8352439 -2.61247830 -1.974666  4 -0.09556612 -0.03370276
## 11  1.6816540  1.681654  2.7204433  2.55538308 -1.974666  2 -1.05645753  1.13975370
## 12  1.6816540  1.681654  2.7204433  2.55538308 -1.974666  3  0.28112242  0.41634234
## 13  1.6816540  1.681654  2.7204433  2.55538308 -1.974666  3  0.41949239 -1.68056035
## 14  1.6816540  1.681654  2.7204433  2.55538308 -1.974666  3  0.09257748  0.38125445
## 15  1.6816540  1.681654  2.7204433  2.55538308 -1.974666  3 -1.38935423  1.58018398
## 16  1.6816540  1.681654  2.7204433  2.55538308 -1.974666  2  0.77630068 -0.34149231
## 17  1.6816540  1.681654  2.7204433  2.55538308 -1.974666  1 -1.59263316 -0.41796446
## 18  1.6816540  1.681654  2.7204433  2.55538308 -1.974666  3  1.41731859 -0.93840191
## 19  1.6816540  1.681654  2.7204433  2.55538308 -1.974666  5 -1.21356119 -0.14223347
## 20  1.6816540  1.681654  2.7204433  2.55538308 -1.974666  5 -0.17774317 -0.16957079
## 21  1.6816540  2.720443  2.7204433  2.55538308 -1.974666  2  1.54240090 -0.84967793
## 22  1.6816540  2.720443  2.7204433  2.55538308 -1.974666  3 -0.07521427 -2.65235730
## 23  1.6816540  2.720443  2.7204433  2.55538308 -1.974666  5  0.66992035 -0.30372825
## 24  1.6816540  2.720443  2.7204433  2.55538308 -1.974666  2 -0.71040015  0.10747575
## 25  1.6816540  2.720443  2.7204433  2.55538308 -1.974666  3 -1.44898507  1.05944755
## 26  1.6816540  2.720443  2.7204433  2.55538308 -1.974666  3  2.59491818  0.45685900
## 27  1.6816540  2.720443  2.7204433  2.55538308 -1.974666  5  1.86745161 -0.32400630
## 28  1.6816540  2.720443  2.7204433  2.55538308 -1.974666  3 -0.26196791  0.31878757
## 29  1.6816540  2.720443  2.7204433  2.55538308 -1.974666  4  0.52161222  1.40719252
## 30  1.6816540  2.720443  2.7204433  2.55538308 -1.974666  5  1.32944107 -0.65495095
```

Finally, we compute the utility provided to each decision makers by each alternative. To do so, we generate measurement error.

```
# computation of the decision makers' utility according to the standardized Gumbel distribution
FD$utility()

# computation of the decision makers' utility according to the discrete uniform distribution
FD$utility("discrete_uniform")

# It is possible to have correlation between alternatives preference (for both student and normal distr
FD$utility("normal", mu=0, sd=2)

# computation of the decision makers' utility according to a student distribution
FD$utility("student", location=0, scale=2, df=4)
```

Here, we take a look at:

```
FD$V # the representative utility
```

```
##    no_choice      good1     good2      good3     good4
## 1          0 11.1270189 12.173397 17.2292940 13.2656231
## 2          0 10.4148116 11.975718 13.0541324  8.4138687
## 3          0 11.1083415  5.284465  4.2998423  7.6896638
## 4          0  7.9938312  7.954862  5.0959428  9.2284277
## 5          0  8.3338438  8.456256  7.3237862  8.0316051
## 6          0  0.6083890  2.293588  4.8575758  0.2981425
## 7          0  6.2255256 10.725417 12.2428306  3.4996757
```

```
## 8           0  4.2150241  6.642560 15.8486668  1.1645365
## 9           0 11.6758827 13.118325 10.2128827  8.7240874
## 10          0  9.5594696 11.006599 10.4492246  7.6054848
## 11          0  3.8681226 12.483035 13.7855279  7.5223724
## 12          0  9.2703452 12.728868 11.1371297 10.5625027
## 13          0  8.6413154  3.000455 -0.7380505  5.5499171
## 14          0  7.2099829 10.397662 12.0550495  2.6991959
## 15          0 12.7297955 20.895619 23.1124207 13.9019731
## 16          0  0.6109729  6.971549  0.4466169  5.2044545
## 17          0  2.2413019  4.535399  7.6393454  1.6921283
## 18          0  6.2246000  6.233861 -0.4269333  3.3567864
## 19          0 11.0003656 11.854024 11.8094144  9.6092298
## 20          0 12.4698415 15.004582 12.5766247 13.3558809
## 21          0  4.3930464  7.407503  0.6928957  3.2452820
## 22          0  8.5308741  1.627499 -1.1277911  6.1208554
## 23          0 12.1871290 12.492156 11.4035078  9.1395033
## 24          0  2.5463605  6.079940  6.8684982  2.6785501
## 25          0  7.9802805 13.707859 19.8569014  9.0506740
## 26          0  7.5397684  9.803483  6.8997193  7.7352357
## 27          0 11.0727625 10.253166  7.9793732 10.1913183
## 28          0  8.8523900 13.880516  9.7623775  7.4808094
## 29          0 12.6203367 15.000894 20.9353343  6.3804735
## 30          0 12.6612824  8.612468  6.5414164  9.8700481
```

```r
FD$Epsilon # the measurement error
```

```
##    no_choice      good1     good2      good3      good4
## 1          0 11.1270189 12.173397 17.2292940 13.2656231
## 2          0 10.4148116 11.975718 13.0541324  8.4138687
## 3          0 11.1083415  5.284465  4.2998423  7.6896638
## 4          0  7.9938312  7.954862  5.0959428  9.2284277
## 5          0  8.3338438  8.456256  7.3237862  8.0316051
## 6          0  0.6083890  2.293588  4.8575758  0.2981425
## 7          0  6.2255256 10.725417 12.2428306  3.4996757
## 8          0  4.2150241  6.642560 15.8486668  1.1645365
## 9          0 11.6758827 13.118325 10.2128827  8.7240874
## 10         0  9.5594696 11.006599 10.4492246  7.6054848
## 11         0  3.8681226 12.483035 13.7855279  7.5223724
## 12         0  9.2703452 12.728868 11.1371297 10.5625027
## 13         0  8.6413154  3.000455 -0.7380505  5.5499171
## 14         0  7.2099829 10.397662 12.0550495  2.6991959
## 15         0 12.7297955 20.895619 23.1124207 13.9019731
## 16         0  0.6109729  6.971549  0.4466169  5.2044545
## 17         0  2.2413019  4.535399  7.6393454  1.6921283
## 18         0  6.2246000  6.233861 -0.4269333  3.3567864
## 19         0 11.0003656 11.854024 11.8094144  9.6092298
## 20         0 12.4698415 15.004582 12.5766247 13.3558809
## 21         0  4.3930464  7.407503  0.6928957  3.2452820
## 22         0  8.5308741  1.627499 -1.1277911  6.1208554
## 23         0 12.1871290 12.492156 11.4035078  9.1395033
## 24         0  2.5463605  6.079940  6.8684982  2.6785501
## 25         0  7.9802805 13.707859 19.8569014  9.0506740
## 26         0  7.5397684  9.803483  6.8997193  7.7352357
## 27         0 11.0727625 10.253166  7.9793732 10.1913183
```

```
## 28          0  8.8523900 13.880516  9.7623775  7.4808094
## 29          0 12.6203367 15.000894 20.9353343  6.3804735
## 30          0 12.6612824  8.612468  6.5414164  9.8700481
```

```
FD$U # the utility of each alternative for each decision maker
```

```
##    no_choice      good1     good2      good3      good4
## 1          0 11.1270189 12.173397 17.2292940 13.2656231
## 2          0 10.4148116 11.975718 13.0541324  8.4138687
## 3          0 11.1083415  5.284465  4.2998423  7.6896638
## 4          0  7.9938312  7.954862  5.0959428  9.2284277
## 5          0  8.3338438  8.456256  7.3237862  8.0316051
## 6          0  0.6083890  2.293588  4.8575758  0.2981425
## 7          0  6.2255256 10.725417 12.2428306  3.4996757
## 8          0  4.2150241  6.642560 15.8486668  1.1645365
## 9          0 11.6758827 13.118325 10.2128827  8.7240874
## 10         0  9.5594696 11.006599 10.4492246  7.6054848
## 11         0  3.8681226 12.483035 13.7855279  7.5223724
## 12         0  9.2703452 12.728868 11.1371297 10.5625027
## 13         0  8.6413154  3.000455 -0.7380505  5.5499171
## 14         0  7.2099829 10.397662 12.0550495  2.6991959
## 15         0 12.7297955 20.895619 23.1124207 13.9019731
## 16         0  0.6109729  6.971549  0.4466169  5.2044545
## 17         0  2.2413019  4.535399  7.6393454  1.6921283
## 18         0  6.2246000  6.233861 -0.4269333  3.3567864
## 19         0 11.0003656 11.854024 11.8094144  9.6092298
## 20         0 12.4698415 15.004582 12.5766247 13.3558809
## 21         0  4.3930464  7.407503  0.6928957  3.2452820
## 22         0  8.5308741  1.627499 -1.1277911  6.1208554
## 23         0 12.1871290 12.492156 11.4035078  9.1395033
## 24         0  2.5463605  6.079940  6.8684982  2.6785501
## 25         0  7.9802805 13.707859 19.8569014  9.0506740
## 26         0  7.5397684  9.803483  6.8997193  7.7352357
## 27         0 11.0727625 10.253166  7.9793732 10.1913183
## 28         0  8.8523900 13.880516  9.7623775  7.4808094
## 29         0 12.6203367 15.000894 20.9353343  6.3804735
## 30         0 12.6612824  8.612468  6.5414164  9.8700481
```

```
FD$choice_order # the order of alternative preference for each decision maker
```

```
##    no_choice good1 good2 good3 good4
## 1          4     5     3     2     1
## 2          4     3     2     5     1
## 3          2     5     3     4     1
## 4          5     2     3     4     1
## 5          3     2     5     4     1
## 6          4     3     2     5     1
## 7          4     3     2     5     1
## 8          4     3     2     5     1
## 9          3     2     4     5     1
## 10         3     4     2     5     1
## 11         4     3     5     2     1
## 12         3     4     5     2     1
```

```
## 13         2     5     3     1     4
## 14         4     3     2     5     1
## 15         4     3     5     2     1
## 16         3     5     2     4     1
## 17         4     3     2     5     1
## 18         3     2     5     1     4
## 19         3     4     2     5     1
## 20         3     5     4     2     1
## 21         3     2     5     4     1
## 22         2     5     3     1     4
## 23         3     2     4     5     1
## 24         4     3     5     2     1
## 25         4     3     5     2     1
## 26         3     5     2     4     1
## 27         2     3     5     4     1
## 28         3     4     2     5     1
## 29         4     3     2     5     1
## 30         2     5     3     4     1
```

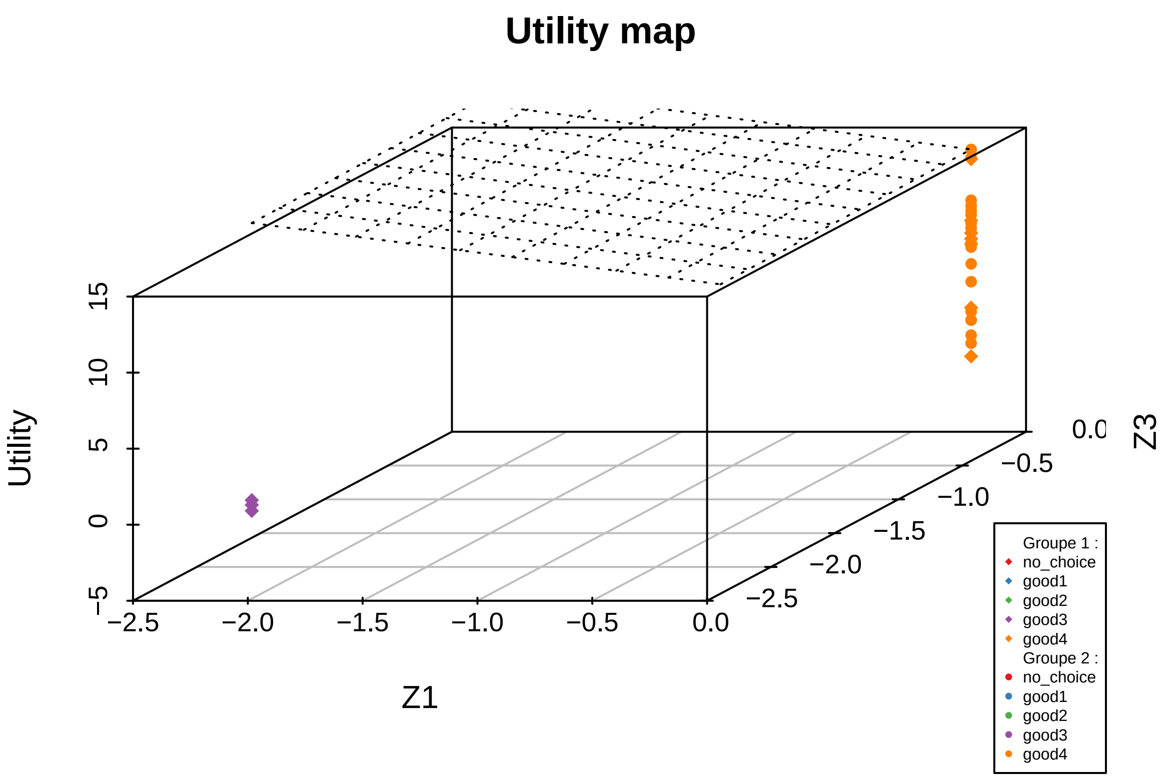
```r
FD$choice # the most usefull alternative for each decision maker
```

```
##    optimal choice
## 1           good4
## 2           good4
## 3           good4
## 4           good4
## 5           good4
## 6           good4
## 7           good4
## 8           good4
## 9           good4
## 10          good4
## 11          good4
## 12          good4
## 13          good3
## 14          good4
## 15          good4
## 16          good4
## 17          good4
## 18          good3
## 19          good4
## 20          good4
## 21          good4
## 22          good3
## 23          good4
## 24          good4
## 25          good4
## 26          good4
## 27          good4
## 28          good4
## 29          good4
## 30          good4
```

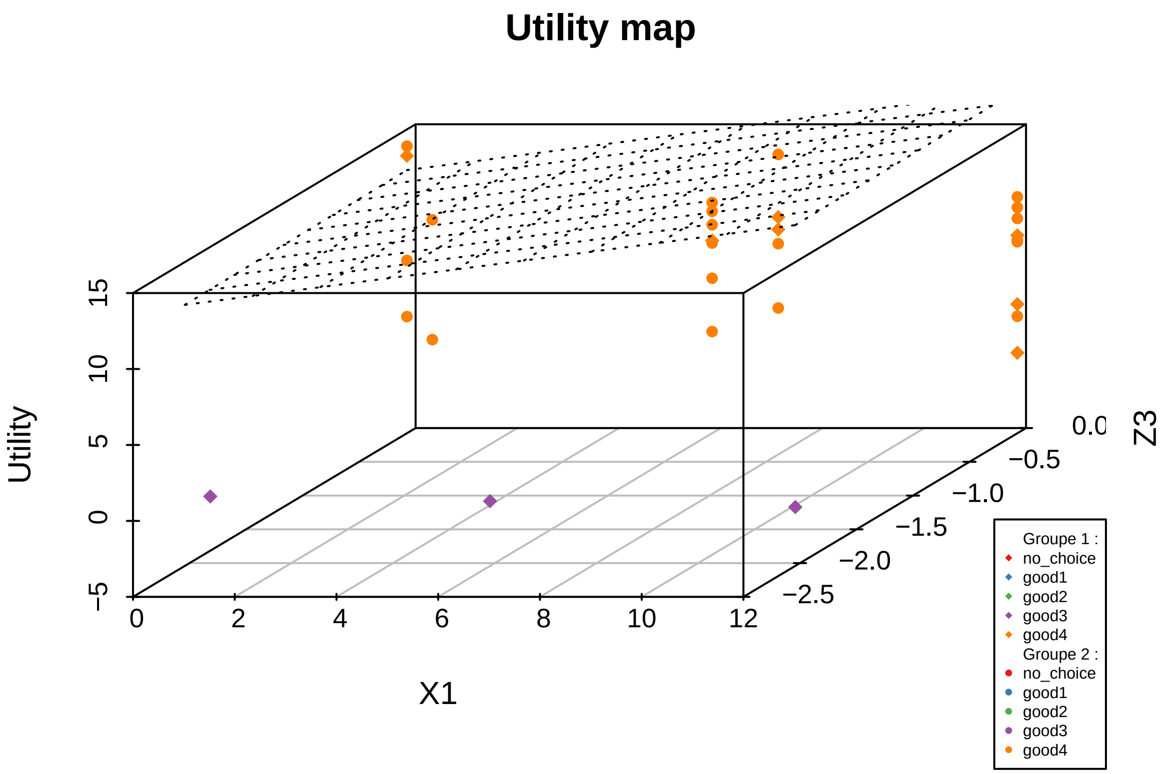
A good advantage of the package RUMdesignSimulator consists in its plot method. The method

\textbf{$map(...)} returns a scatter plot. On this graph, the x-axis, y-axis and z-axis represent the value of two parameters (attributes and characteristics) and the utility provided by the optimal alternative for any decision maker.

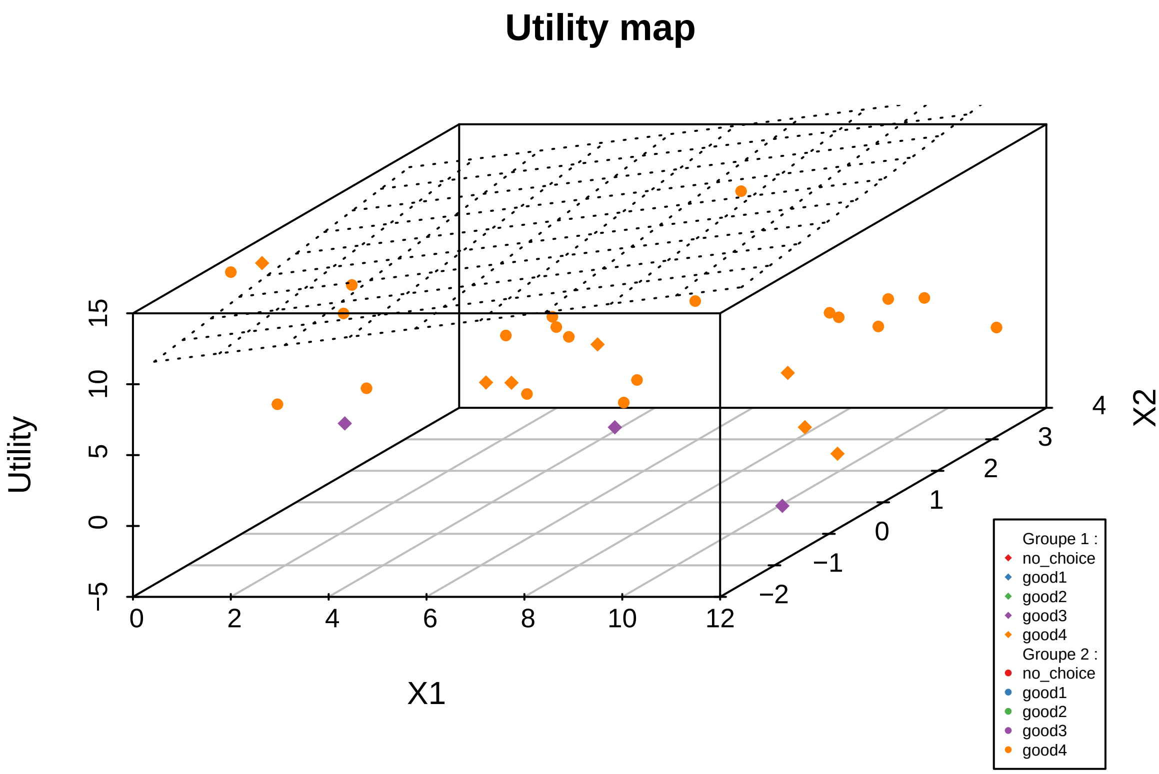
```
# Drawing a 3D preference mapping:

# Map representing the choice of the decision makers and the utility provided by this choice according
FD$map("Z1", "Z3")
```




```
FD$map("X1", "Z3")
```


Utility map
Utility
15
10
5
0
−5
0
2
4
6
8
10
12
X1
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
Z3
Groupe 1 :
no_choice
good1
good2
good3
good4
Groupe 2 :
no_choice
good1
good2
good3
good4

```
FD$map("X1", "X2")
```




```
# Generation of designs:

# generation of the full factorial design with row data
FFD <- FD$design(choice_set_size=2, clustered=0)
```

```
## $Alternatives_names
## [1] "no_choice" "good1"     "good2"     "good3"     "good4"
##
## $choice_set_size
## [1] 2
##
## $number_of_alternatives
## [1] 5
##
## $no_choice
## [1] TRUE
##
## $Decision_Makers_attributes_names
## [1] "X1"          "X2"          "X3"          "I(X1 * X2)"
##
## $Alternatives_attributes_names
## [1] "Z1"       "Z2"       "Z3"       "I(Z1^2)"
##
## $D_score
## [1] 0
##
```

```
## $beta_value
##                value
## Nelder-Mead 113.2746
##
## $beta_hat
##                   X1         X2       X3 I(X1 * X2)       Z1        Z2       Z3  I(Z1^2)
## Nelder-Mead 5.356518 -0.2197439 2.160536 -0.7725891 3.223472 -3.761842 3.474168 3.698656
##
## $mean_real_beta
##          X1          X2          X3  I(X1 * X2)          Z1          Z2          Z3     I(Z1^2)
##  0.43746368  0.70160243  0.96149635  1.19529309 -1.97466590  3.36666667  0.07108933 -0.02632944
```

```r
#by default, name="FuFD", choice_set_size = nb_alternatives
View(FFD)
```

Henceforth, the alternatives, decision makers, preference coefficients and associated utility are entirely setup. In consequence, we can draw experimental designs. Below, we build a full factorial experimental design where the number of alternatives within each choice set is 2. Moreover, the attributes and characteristics are not treated.

Often, the data is treated. Sometimes, the data is clustered. The number of each cluster is called *level*. Furthermore, the clusters are formed by running k-means algorithms. Finally, after clustering, the new value of an attribute or a characteristic is the average of its cluster.

```r
FFD <- FD$design(name="FuFD",choice_set_size=2, clustered=1, nb_levels_DM=c(3, 3, 4, 2), nb_levels_AT=c
```

```
## $Decision_makers_duplicates
##  [1] "11" "15" "16" "17" "20" "24" "25" "27" "29" "30"
##
## $Alternatives_names
## [1] "no_choice" "good1"     "good2"     "good3"     "good4"
##
## $choice_set_size
## [1] 2
##
## $number_of_alternatives
## [1] 5
##
## $no_choice
## [1] TRUE
##
## $Decision_Makers_attributes_names
## [1] "X1"         "X2"         "X3"         "I(X1 * X2)"
##
## $Alternatives_attributes_names
## [1] "Z1"      "Z2"      "Z3"      "I(Z1^2)"
##
## $D_score
## [1] 0
##
## $beta_value
##                value
## Nelder-Mead 110.5749
##
```

```
## $beta_hat
##                      X1        X2       X3 I(X1 * X2)       Z1        Z2        Z3  I(Z1^2)
## Nelder-Mead -0.1538202 -3.052899 1.191764   7.904376 2.828883 0.5916045 -7.102658 3.904312
##
## $mean_real_beta
##          X1          X2          X3  I(X1 * X2)          Z1          Z2          Z3     I(Z1^2)
##  0.43746368  0.70160243  0.96149635  1.19529309 -1.97466590  3.36666667  0.07108933 -0.02632944
```

In addition, after clustering, the new value of an attribute or a characteristic may be the numero of its cluster. This is done by defining **clustered=2**.

```
# generation of the full factorial design with categorial data
FFD1 <- FD$design(choice_set_size=2, clustered=2, nb_levels_DM=c(2, 3, 4, 2), nb_levels_AT=c(2, 2, 2, 2
```

```
## $Decision_makers_duplicates
##  [1] "7"  "9"  "11" "15" "16" "17" "20" "24" "25" "26" "27" "29" "30"
##
## $Alternatives_names
## [1] "no_choice" "good1"     "good2"     "good3"     "good4"
##
## $choice_set_size
## [1] 2
##
## $number_of_alternatives
## [1] 5
##
## $no_choice
## [1] TRUE
##
## $Decision_Makers_attributes_names
## [1] "X1"         "X2"         "X3"         "I(X1 * X2)"
##
## $Alternatives_attributes_names
## [1] "Z1"      "Z2"      "Z3"      "I(Z1^2)"
##
## $D_score
## [1] 0
##
## $beta_value
##                value
## Nelder-Mead 108.3178
##
## $beta_hat
##                    X1      X2        X3 I(X1 * X2)      Z1       Z2        Z3  I(Z1^2)
## Nelder-Mead -81.11869 208.996 -39.68008  -68.16502 434.319 41.52509 -362.4037 130.1612
##
## $mean_real_beta
##          X1          X2          X3  I(X1 * X2)          Z1          Z2          Z3     I(Z1^2)
##  0.43746368  0.70160243  0.96149635  1.19529309 -1.97466590  3.36666667  0.07108933 -0.02632944
```

Unfortunately, the number of questions asked to each decision maker is most of the time too big to be realistic. In consequence, only a random subset of questions can be asked to the decision makers. The result is a *random fractional factorial design*. The number of question asked to each decision maker is \textbf{nb_question=2}.

```r
FFD2 <- FD$design(name="FrFD", choice_set_size=2, clustered=2, nb_levels_DM=c(2, 3, 4, 2), nb_levels_AT=
```

```
## $Decision_makers_duplicates
##  [1] "11" "15" "16" "17" "20" "21" "24" "25" "27" "29" "30"
##
## $Alternatives_names
## [1] "no_choice" "good1"     "good2"     "good3"     "good4"
##
## $choice_set_size
## [1] 2
##
## $number_of_alternatives
## [1] 5
##
## $no_choice
## [1] TRUE
##
## $Decision_Makers_attributes_names
## [1] "X1"         "X2"         "X3"         "I(X1 * X2)"
##
## $Alternatives_attributes_names
## [1] "Z1"      "Z2"      "Z3"      "I(Z1^2)"
##
## $D_score
## [1] 0
##
## $beta_value
##                value
## Nelder-Mead 37.71471
##
## $beta_hat
##                    X1       X2        X3 I(X1 * X2)      Z1       Z2       Z3   I(Z1^2)
## Nelder-Mead -754.1425 539.2107 -105.6733   628.1675 276.159 364.4114 589.2687 -686.2568
##
## $mean_real_beta
##          X1          X2          X3  I(X1 * X2)          Z1          Z2          Z3     I(Z1^2)
##  0.43746368  0.70160243  0.96149635  1.19529309 -1.97466590  3.36666667  0.07108933 -0.02632944
```

Yet, we want to express this design in wide format.

```r
FFD3 <- FD$design(name="FrFD", choice_set_size=2, clustered=2, nb_levels_DM=c(2, 3, 4, 2), nb_levels_AT=
```

```
## $Decision_makers_duplicates
##  [1] "5"  "7"  "9"  "11" "12" "14" "15" "16" "17" "18" "20" "21" "22" "24" "25" "26" "27" "28" "29"
##
## $Alternatives_names
## [1] "no_choice" "good1"     "good2"     "good3"     "good4"
##
## $choice_set_size
## [1] 2
##
## $number_of_alternatives
```

```
## [1] 5
##
## $no_choice
## [1] TRUE
##
## $Decision_Makers_attributes_names
## [1] "X1"         "X2"         "X3"         "I(X1 * X2)"
##
## $Alternatives_attributes_names
## [1] "Z1"      "Z2"      "Z3"      "I(Z1^2)"
##
## $D_score
## [1] 0
##
## $beta_value
##                value
## Nelder-Mead 36.30784
##
## $beta_hat
##                   X1        X2        X3 I(X1 * X2)        Z1      Z2        Z3  I(Z1^2)
## Nelder-Mead 9.587152 -4.247282 0.8484649   3.823091 -5.676396 4.35986 0.3943429 2.546505
##
## $mean_real_beta
##          X1          X2          X3  I(X1 * X2)          Z1          Z2          Z3     I(Z1^2)
##  0.43746368  0.70160243  0.96149635  1.19529309 -1.97466590  3.36666667  0.07108933 -0.02632944
```

Finally, a small summary function calls some elements back.

```
summary.Exepriment(FD) # a summary of the experimental design
```

```
## Alternatives’ names: no_choice good1 good2 good3 good4
## Attributes’ alternatives’ names: Z1 Z2 Z3 I(Z1^2)
## Groups of Decision makers: 10 20
## Decision Makers’ characteristics’ names: X1 X2 X3 I(X1 * X2)
```

Developpement

Files

- The file **tutorial.Rmd** contains the tutorial above for reproductibility.
- The folder **R** contains the R files. Inside this folder, the file **experiment.R** is the main file. Moreover, the utility of the other files is explicitly given by their name.
- The folder **man** contains the documentation about the functions of the package

Adding new features

Some user may need more tools than the actual ones. Anticipating future needs, we organized the R files so that only one file need to be altered.
To add new distributions: * open the file distribution.R * add a new distribution * reference the new distribution into the function *generation*, give it a relevant name for calling

To add new designs: * open the file designs.R * implement a new design * reference the new design into the function *call_design*, give it a relevant name for calling

For more information, do not hesitate to contact me at antoine.dubois.fr@gmail.com