
A STEP-BY-STEP GUIDE TO DESIGN, IMPLEMENT, AND ANALYZE A DISCRETE CHOICE EXPERIMENT

Daniel Pérez-Troncoso
Faculty of Economics
University of Granada
danielperez@ugr.es

September 24, 2020

ABSTRACT

Discrete Choice Experiments (DCE) have been widely used in health economics, environmental valuation, and other disciplines. However, there is a lack of resources disclosing the whole procedure of carrying out a DCE. This document aims to assist anyone wishing to use the power of DCEs to understand people's behavior by providing a comprehensive guide to the procedure. This guide contains all the code needed to design, implement, and analyze a DCE using only free software.

1 Introduction

Discrete choice experiments (DCE) have been widely used in health economics [1–4], environmental valuation [5, 6], and other disciplines where there is not a market institution where consumers' decision can be observed. For addressing the lack of information, in a DCE respondents are presented with different alternatives to choose from (simulating a market situation). Each alternative comprises several attributes and levels to force consumers to make trade-offs according to their preferences [7, 8]. Despite the usefulness of discrete choice experiments, they can be hard to design, implement, and analyze, and usually paid software is recommended to carry out the research. This document is aimed to elaborate on the full process of studying consumers' preferences through a discrete choice experiment.

2 Elements in a discrete choice experiment

Before starting with the steps of the procedure, it is worth to understand the parts and nomenclature used to design the elements of a DCE.

A discrete choice experiment is usually administered through a questionnaire than contains [at least] three parts: a) an introduction, 2) the DCE itself, 3) respondent information [1]. The introduction is necessary to make respondents understand what they are responding to, why are they doing that, and how to do it correctly. The second section, the most important one, contains the DCE itself. In this section, respondents repeat a choice task (Example in Table 1) between different hypothetical scenarios as many times as the analyst decides. These choice tasks are known as 'choice sets', and they usually contain 2 or 3 alternatives. Attributes and levels are the main components of the choice sets. Attributes are the categories where the characteristics of the hypothetical product are classified (see the left column in Table 1). Levels are the different functionalities of the product classified by attributes. Following the example in Table 1, a possible selection of attributes and levels is presented in Table 2.

In the third section, respondents can be asked about anything relevant to the analysis of their decisions. Usually, this section includes socio-demographic questions (like income, age, education, and gender).

If the attributes and levels in Table 2 were combined to create all the possible choice sets, they would lead to $3^2 \times 2^2 = 108$ different profiles. Pairing the profiles we would have a total of $(108 \times 107)/2 = 5778$ different choice sets. A DCE using all the existing combinations is known as a full factorial design. However, most cases we do not have such a large sample, so "the central question is then how to combine the alternatives from the full factorial design

Table 1: Example Choice Set

Attributes	Option 1	Option 2	Option 3
Efficacy	100%	80%	
Side effects	nausea in 1 out of 10,000 patients.	none	
Dose	Once a week	Once a day	
Administration	Injection	Oral	
Price	150€/month	200€/month	0€
	Choose option 1	Choose option 2	Choose option 3

Table 2: Example selection of attributes and levels

Attributes	Levels	No. of levels
Efficacy	1) 80%, 2) 90%, 3) 100%	3
Side effects	1) nausea in 1 out of 10,000 patients, 2) none	2
Dose	1) Once a day, 2) Once a week, 3) Once a month	3
Administration	1) Oral, 2) Injection	2
Price	1) 100€/month, 2) 150€/month, 3) 200€/month	3

into the choice sets so that a maximum amount of information is extracted [...] [9, p. 284]. In this document, we will address an optimization technique to obtain the maximum possible information from our design.

3 Attributes and levels selection

Attributes and levels selection is an essential part of the DCE design. Poor selection will lead to invalid results, so it is worth doing extensive qualitative research before making the final selection. Some useful references are the WHO guide [2], the Gerard et al. manual [4], and the ISPOR checklist [10]. A literature review will be essential but can be supplemented by consultation with experts and stakeholders. A common strategy here is to gather all possible attributes and levels into one long list and review it by discarding some and merging others. As a general rule, the selection of attributes and levels should be justifiable by evidence. As the number of attributes and levels increases, each response provides less information, so a balance needs to be found between specification and efficiency. Besides, a limited selection of attributes and levels is recommended so as not to confuse the respondent. 70% of existing DCEs use from 3 to 7 attributes [1].

Conversely, an incomplete specification of attributes and levels could lead to a large estimation error (note that the results of this methodology will be analyzed in the random utility framework [11]). According to this theory, the utility that an individual n gets from a product j is given by a deterministic component, V_{nj} , and a random component not observed by the researcher, ϵ_{nj} .

$$u_{nj} = V_{nj} + \epsilon_{nj} \quad (1)$$

In Equation 1, the deterministic component, V_{nj} , is composed of the attributes' levels and its coefficients, $\beta_{nj}x_{nj}$. As higher the number of attributes and levels is, the lower the error component, ϵ_{nj} and vice versa. That is why we must balance simplicity and precision.

Finally, when deciding the attributes and levels in a DCE we have to include a price attribute with two or more cost or price levels. This attribute is vital, as respondents will make most decisions with it in mind. It will allow us to measure the trade-offs between functionalities (levels) and monetary units. Thus, we can measure willingness to pay (WTP) as a simple division of the level coefficient by the price attribute coefficient.

4 Pilot study and experimental design

The efficient design of a DCE requires prior information in the form of "prior coefficients" of the model we want to estimate to perform an optimization procedure. This procedure consists of creating a design matrix (experimental design) that reduces the variance of the coefficients that we are going to calculate [9]. One of the most common ways of

estimating this variance is

$$d - error = |\Omega^{1/k}| \quad (2)$$

where Ω is the covariance matrix of the model coefficients, and K is the number of parameters of the model. Thus, an algorithm can be employed to iterate random designs matrices until it finds the experimental design with lower D-error. This procedure ensures the efficiency of the design and the accuracy of the estimators.

A pilot study consists of creating an efficient design based on a set of prior coefficients equal to zero and presenting the resulting design to a limited number of respondents to achieve the first set of actual prior coefficients. Once we achieved the coefficients, we can use them to generate another efficient design.

The first step is to design a pilot DCE with priors equal to zero. To do it, we will use R (<https://www.r-project.org/>), (optional but recommended) RStudio (<https://rstudio.com/>), and the package 'idefix' [17].

```
1. install.packages("idefix")
2. library(idefix)
3. levels <- c(3,2,3,2,3)
4. coding <-c("E", "E", "E", "E", "E")
```

The first and second line will install and activate the package that we are going to use to create an efficient design. At line 3, we are creating a vector where each element is an attribute, and its value is the number of levels in that attribute. At line 4, we specify another vector with the type of coding that we are going to use in each attribute. In this case, we will use "effects coding" so we write an "E" per attribute. If we want to see the different alternatives from all attributes and levels combination, we can create and display the profiles:

```
5. Profiles (lvls=levels, coding=coding)
```

The input at line 5 will generate all different alternatives from our design. In this case, we can see that 108 profiles were generated: $3^3 * 2^2 = 108$.

Next, we want to generate the D-efficient design. To do so, we will use Fedorov modified algorithm [13] included in the -idefix- package. This algorithm generates a random initial design from the set of profiles and randomly switches the levels and compares the D-error [14]. This process goes on until the iteration $n - 1$ and n exhibit the same D-error. By reducing D-error we are getting close to the principles of good DCE design: orthogonality, level balance, minimal overlap, and utility balance [15].

As we mentioned before, because this is the first pretest, we do not have any prior information. Thus, when specifying the vector with the prior coefficients, we need to set them all to zero.

```
6. priors <- c(0, 0, 0, 0, 0, 0, 0, 0, 0)
7. s <- diag(length(priors))
8. sim <- MASS::mvrnorm(n = 500, mu = priors, Sigma = s)
9. sim <- list(sim[, 1:1], sim[, 2:9])
10. alt.cte <- c(0, 0, 1)
11. d<-CEA(lvls=levels, coding=coding, n.alts=3, n.sets=16, alt.cte=alt.cte,
    par.draws=sim, no.choice=TRUE, best=TRUE)
```

At line 6, we have a total of 9 zeros, although we have 13 coefficients. That is because we are omitting a level per attribute (due to the effects coding) and adding an additional zero as the coefficient of the dummy representing the opt-out alternative. Thus, the amount of coefficients in the priors vector is $(l - k) + 1$ where l is the number of levels and k is the number of attributes.

Lines 7 and 8 are part of a simulation procedure where 500 random draws are obtained from a normal distribution with mean equal to the priors specified. Those results will be used to simulate a response pattern and generate a design that reduces the estimators' variance. At line 9, we create two lists, the first one for the coefficient of the dummy variable in the opt-out alternative, and the other for the rest of the coefficients. At line 10, we create a vector indicating the position of the opt-out alternative and, finally, line 11 produces an output in d with the D-efficient design. As we can see, we are creating a D-efficient design based on the priors stored in $priors = (0, 0, 0, 0, 0, 0, 0, 0, 0)$, with five attributes with, respectively, 3, 2, 3, 2, 3 levels. We are using effects coding, three alternatives per choice set (the third one is a constant no-choice/opt-out), and sixteen choice sets per respondent.

```
12. design <- d$design
```

Table 3: Dummy vs. effects coding

	Dummy coding			Effects coding		
	β_2	β_3	β_4	β_1	β_2	β_3
Level 1	0	0	0	1	0	0
Level 2	1	0	0	0	1	0
Level 3	0	1	0	0	0	1
Level 4	0	0	1	-1	-1	-1

Figure 1: Screenshot to design matrix in R

```

~/ ↵
      alt3.cte var11 var12 var21 var31 var32 var41 var51 var52
set1.alt1    0     1     0     1    -1    -1    -1    -1    -1
set1.alt2    0    -1    -1    -1     1     0     1    -1    -1
no.choice    1     0     0     0     0     0     0     0     0
set2.alt1    0     1     0     1     1     0    -1    -1    -1
set2.alt2    0     0     1    -1     0     1     1     1     0
no.choice    1     0     0     0     0     0     0     0     0

```

At line 12, we store the design matrix in the variable 'design'. Once we have the design matrix, we need to interpret it to obtain the choice sets. Our design matrix is an $n \times m$ matrix where n is the number of choice sets times the number of alternatives, and m is equal to the number of coefficients in the priors vector, $(l - k) + 1$. In this design, we are using 'effects coding' where all levels are written as dummy variables except for one omitted level.

Now we want to interpret the design matrix to understand how alternatives are distributed in across the choice sets. In Table 3 we can see how, depending on the coding used, the variables take one value or another to represent one attribute's level.

For example, figure one shows the output of *design*. In this case, the choice set one has alternative 1 with levels (1, 1, 3, 2, 3) vs. alternative 2 with levels (3, 2, 1, 1, 3) versus the "none of them option". The second choice set contains alternative 1, (1, 1, 1, 2, 3), and alternative 2, (2, 2, 2, 1, 1).

5 Survey design

DCEs are commonly administered via online surveys. In this section, we will focus on how to design an online survey. Our survey will be divided into three parts: 1) introduction, 2) DCE, and 3) respondent information. For this example, in the introduction, we will add an introductory text, explaining to the respondents how the experiment works, and a timer that will not let them skip the introduction until after a minute. To do this, we will create two HTML documents with the following code:

```

<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <title>Title in the tab</title>

    <script src="http://code.jquery.com/jquery-latest.min.js" type="text/javascript"></script>
    <link rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"

```

```

    integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwA0H8WgZl5MYxXfFc+NcPb1dKGj7Sk"
    crossorigin="anonymous">
</head>

<body>
  <h1>Welcome to my survey</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
    ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation
    ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in
    reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur
    sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id
    est laborum.</p>

  <p>Before you go to the survey, you need to wait 1 minute so please use this time to read the
    introductory text<p></p>
  <button id="exit" onclick="document.location='redirect.html'">Go to survey!</button>

  <script>
    $(document).ready(function() {
      $("#exit").prop("disabled", true);
      setTimeout(function(){$("#exit").prop("disabled", false)}, 60000);
    });
  </script>
</body>
</html>

```

The code above can be copied and pasted into a plain text editor and saved as .html. The dummy text in `< p >< /p >` can be modified according to the necessities of the study, same with the 60000 value (equivalent to 1 minute) in the `setTimeout` function.

The link on the "Go to Survey" button redirects to another .html file whose function is to send the respondent randomly to an A or B link. Thus, the code below will be useful when the survey is blocked into two parts and we want to send respondents randomly to block A or B.

```

<html>
<head>
  <title></title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <script type="text/javascript">
    var pageArr = ["https://cran.r-project.org/", "http://gretl.sourceforge.net/"];
    document.location.href = pageArr[Math.ceil(Math.random()*2)-1];
  </script>
</head>

<body>

</body>
</html>

```

This code needs to be saved and named `redirect.html`. In the example code, the page randomly redirects to either the R page or the Gretl page. Those links can be modified with the links of both parts of the survey to create the random assignment. The introduction and the redirect pages need to be uploaded to the Internet as a new webpage or part of an existing domain.

Next, we will design the blocks in Google Forms. The optimal option here is Qualtrics, but this is a paid software which only some universities and companies have access too. Although Google Forms has many limitations, we can expect it to remain always free. Thus, we will create two Google Forms, one for each block. In each form we need to:

- click on Settings / Presentation / Shuffle question order,
- create two sections,
- create the socio-demographic questions in section 2,
- create the choice sets in section 1.

To create the choice sets best option might be using 'multiple choice' questions and adding an image to each question. That image can be designed at any text editor and could be something similar to Table 2 (icons can be added to improve respondent's interpretation).

6 Analysis of results

The results of our discrete choice experiment will be analysed in line with McFadden's Random Utility Theory (RUT) [11]. According to RUT, the utility that an individual n gets from an alternative j , u_{nj} , is given by a deterministic component, V_{nj} , plus an unobserved component, ϵ_{nj} [19]:

$$u_{nj} = V_{nj} + \epsilon_{nj} \quad (3)$$

Additionally, the deterministic component of the utility, V_{nj} , can be specified as the observed characteristics of the alternatives (levels), x_{nj} , related to the value that respondents assign to each level, β_{nj} . We can specify this relationship as

$$u_{nj} = \beta_{nj}x_{nj} + \epsilon_{nj}, \quad (4)$$

where the value of the betas can be estimated according to different assumptions. For instance, if we assume that there is not taste heterogeneity among respondents, the equation can be specified as

$$u_j = \beta x_j + \epsilon_j \quad (5)$$

where the utility does not vary among respondents but does vary among alternatives (note the deletion of the sub-index n). This model is known as the conditional logit model (CLM) [18], and it results in the same coefficients for all respondents. While the CLM is the easy and reliable model recommended by the ISPOR task force report [18], it has strong assumptions (IIA) and some limitations (it assumes preference homogeneity and ignores the panel nature of the data i.e. do not consider the 'personid' variable.)

To avoid the limitations of the CLM, we might want to take into account taste heterogeneity among consumers. In this case, the utility will be specified as

$$u_{nj} = \beta_{nj}x_j + \epsilon_{nj} \quad (6)$$

where we compute a coefficient per respondent (note that only x_j lacks the n sub-index because all respondents are presented with the same levels). This specification can be estimated with the mixed logit model (XLM). The XLM is a flexible model adaptable to any choice situation [19]. In this model, the coefficients are computed per respondent over a simulated density $f(\beta)$. Thus we are computing something similar to a CLM per respondent. When the model converges, it reports the mean of the β s of each level along with its standard deviation. Thus, we can know whether or not differences among respondents parameters are significant.

6.1 Reshaping the data

We need to resize the data from the Google Forms survey. When we download the survey data from Google Forms, the data is shaped in 'wide' format (Table 4), and we need to resize it to 'long' format (Table 5) for the logit model to understand it. Since we have blocked the survey into two parts, we will do this individually for each block. To do this, we will use the following commands.

```
12. data<-data[2:ncol(data)]
13. personid <- rownames(data)
14. personid <- as.integer(personid)
15. data <- cbind(personid, data)
```

First, we remove the "timestamp" variable (12), then we need to create a person id (13, 14, 15). At this point, we have our dataset as in Table 4 where columns represent the different choice sets. It is convenient having the same id for the choice sets than in the design matrix (for example, column 1 in Table 4 is the response of respondent 1 to set 1 in Figure 1). Then, the library `-reshape2-` will help us to switch from 'wide' to 'long' shaped format (16, 17, 18).

Table 4: Wide shaped data set

	1	2	3	4	5
1	A	B	A	C	B
2	B	C	A	B	B
3	C	C	A	A	C
4	A	A	C	B	A
...
N	A	B	B	C	C

Choice sets in columns, responses in rows.

Table 5: Long shaped data set

personid	cs	choice	alt	...	price
1	1	1	1	...	150
1	1	0	2	...	100
1	1	0	3	...	200
1	2	0	1	...	200
1	2	1	2	...	100
1	2	0	3	...	150
...
50	800	0	3	...	100

Respondent id, choice set id, choice dummy, alternative id, and alternative specific variables (levels) in columns. For example, respondent 1 chooses alternative 1 (option A) in choice set 1.

```
16. install.packages("reshape2")
17. library(reshape2)
18. data <- melt(data, id.vars=c("personid"))
```

Once we have our dataset reshaped, we need a row per alternative. To do this, we will multiply the number of choice sets (which, at this point, coincides with the number of rows per respondent) per the number of alternatives. In our case, we have three alternatives per choice set (19, 20). We also need an alternative id (21, 22, 23) and a choice set unique id (24, 25, 26).

```
19. data <- rbind(data, data, data)
20. data <- data[order(data$personid, data$variable),]
21. x <- nrow(data)/3
22. alt <- rep(1:3, x)
23. data <- cbind(data, alt)
24. cs <- rep(1:x, each= 3)
25. cs <- sort(cs)
26. data <- cbind(data,cs)
```

Next, the package `dplyr` is needed for the final steps (27, 28). At line 29, we are creating the choice variable that will take value 1 if the alternative in its row is selected or 0 otherwise. This command only works if the response variable was coded as "A", "B", "C" for options A, B, C. If not, it has to be modified for it to work.

```
27. install.packages("dplyr")
28. library(dplyr)
29. data <- mutate(data, choice=ifelse(value == "A" & alt=="1" | value=="B" & alt=="2"
  | value=="C" & alt=="3", 1, 0))
```

Finally, we have to add the alternative specific variables directly from our design matrix. First, we need to select the part of the design matrix that correspond to our block. For instance, if we are working with Block 1 with choice sets 1-8, we want to split the design matrix into two parts (cs 1-8, 9-16)¹.

```
30. design1 <- [1:24,]
31. design2 <- [25:48,]
```

Thus, 'design1' contains the first 8 choice sets while 'design2' contains from the choice set 9 to the choice set 16. To add the alternative specific variables to each response first, we adapt the 'design1(2)' to the number of responses (line 32), and then we merge responses and design (line 33).

¹The choice sets id in the database must correspond with the choice sets id in the design matrix. For example, responses in column '1' of Table 4 need to be responses to choice set '1' of the design matrix. Once data has been reshaped from wide to long format, we can bind the response data (with $8 \times 3 = 24$ rows per respondent) to the design matrix (which divided into two groups has 24 rows). Thus, responses to choice sets 1-8 will be binded to the first 24 rows of the design matrix while responses to choice sets 9-16 will be binded to the last 24 rows of the design matrix.

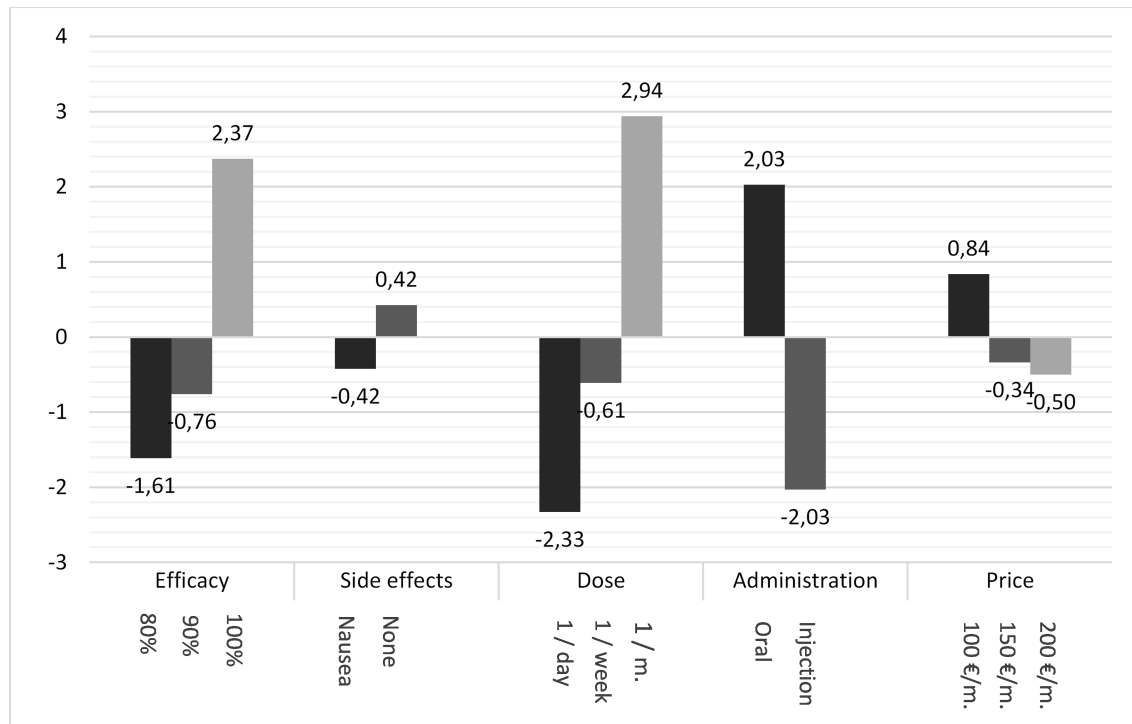


Figure 2: CLM results

```
32. design1 <- design1[rep(seq_len(nrow(design1)), length(personid)), ]
33. final1<- cbind(data, design1)
```

Once this process has been carried out in both parts of the design (final1 and final2), we can merge both datasets:

```
34. final <- rbind(final1, final2)
```

6.2 Computing and reporting the CLM

Now we can estimate the conditional logit model. Typing lines 33, 34, and 35, we will have enough information to report the results as in Table 6. As we can see from the syntax at line 34, the model output does not contain the coefficient of omitted levels (100% efficacy, no side effects, etc.). However, we can compute it as the negative sum of the rest of the coefficients in the same attribute.

```
33. library(survival)
34. resultsCLM <- clogit(choice~alt3.cte + Var11 + Var12 + Var21 + Var31 + Var32 +
  Var41 + Var51 + Var52 + strata (cs), data(final)
35. summary(resultsCLM)
```

In the CLM, we cannot include respondents characteristics since they do not vary among alternatives. The coefficients that we obtain are associated with the effect (or 'weight') of the level in the decision making process in relation to the mean attribute effect [18]. We can improve the interpretation of the conditional logit model by plotting the coefficients of the regression as done in Figure 2. For example, in this case respondents obtain less utility from increasing efficacy from 90% to 80% (+0.85) than from 90% to 100% (+3.13). However, the most important attribute is 'Dose' because the difference between its lowest and highest level is greater than the difference in 'Efficacy' (5.27 vs. 3.98).

6.3 Computing and reporting the XLM

To estimate the XLM, we need to set up our data set by typing the following code:

```
36. D <- dfidx(final, choice="choice", idx = list(c("cs", "personid"), "alt"), idnames
```

Table 6: Results of the CLM

Attributes	Levels	Coefficient	SE	p-value
Efficacy	80%	-1.61188	0.11955	p<0.0000
	90%	-0.75922	0.08631	p<0.0000
	100%	2.3711	(Omitted level)	
Side effects	Nausea 1/10,000 patients	-0.42223	0.06407	p<0.0000
	None	0.42223	(Omitted level)	
Dose	Once a day	-2.32870	0.14545	p<0.0000
	Once a week	-0.61081	0.08389	p<0.0000
	Once a month	2.93951	(Omitted level)	
Administration	Oral	2.02899	0.11731	p<0.0000
	Injection	-2.02899	(Omitted level)	
Price	100€/month	0.83695	0.08754	p<0.0000
	150€/month	-0.33681	0.09101	p<0.0000
	200€/month	-0.50014	(Omitted level)	

```
= c("cs", "alt")
```

```
37. resultsXLM <- mlogit(choice ~ alt3.cte + Var11 + Var12 + Var21 + Var31 + Var32 +
  Var41 + Var51 + Var52 | 0, data=D, rpar=c(Var11="n", Var12="n", Var21="n",
  Var31="n", Var32="n", Var41="n", Var51="n", Var52="n"), R=100, halton=NA, panel=TRUE)
```

At line 36 we are shaping the data for the `-mlogit-` function to read it properly (using the same variable names that we used along this document should work). At line 37 we are computing the XLM; the `-rpar-` option allow us to introduce a vector with the coefficients that we want to be random and its distribution ("n" for normal).

Unlike the CLM, the XLM reports the standard deviation of the coefficients. If the standard deviation is significant, there is taste heterogeneity among respondents.

References

- [1] F. R. Johnson et al. "Constructing experimental designs for discrete-choice experiments: Report of the ISPOR conjoint analysis experimental design good research practices task force". *Value Heal.*, vol. 16, no. 1, pp. 3–13, Jan. 2013, doi: 10.1016/j.jval.2012.08.2223.
- [2] WHO. "How to conduct a discrete choice experiment for health workforce recruitment and retention in remote and rural areas," 2012. [Online], Available: <https://www.who.int/hrh/resources/dceguide/en/>.
- [3] E. Lancsar and J. Louviere. "Conducting Discrete Choice Experiments to Inform Healthcare Decision Making," *Pharmacoeconomics*, vol. 26, no. 8, pp. 661–677, 2008, doi: 10.2165/00019053-200826080-00004.
- [4] K. Gerard, M. Ryan, and M. Amaya-amaya, "Conducting Discrete Choice Experiments to Inform Healthcare Decision Making," *Using Discrete Choice Experiments to Value Health and Health Care*, 2008.
- [5] D. Hoyos, "The state of the art of environmental valuation with discrete choice experiments," *Ecological Economics*, vol. 69, no. 8. Elsevier, pp. 1595–1603, Jun. 15, 2010, doi: 10.1016/j.ecolecon.2010.04.011.
- [6] G. Ewing and E. Sarigöllü, "Assessing Consumer Preferences for Clean-Fuel Vehicles: A Discrete Choice Experiment," *EJ. Public Policy Mark.*, vol. 19, no. 1, pp. 106–118, Apr. 2000, doi: 10.1509/jppm.19.1.106.16946.
- [7] M. Espinosa-Goded, J. Barreiro-Hurlé, and E. Ruto, "What do farmers want from agri-environmental scheme design? A choice experiment approach," *J. Agric. Econ.*, vol. 61, no. 2, pp. 259–273, Jun. 2010, doi: 10.1111/j.1477-9552.2010.00244.x.
- [8] M. K. Hidrue, G. R. Parsons, W. Kempton, and M. P. Gardner, "Willingness to pay for electric vehicles and their attributes," *Resour. Energy Econ.*, vol. 33, no. 3, pp. 686–705, Sep. 2011, doi: 10.1016/j.reseneeco.2011.02.002.
- [9] F. Carlsson and P. Martinsson, "Design techniques for stated preference methods in health economics," *Health Econ.*, vol. 12, no. 4, pp. 281–294, Apr. 2003, doi: 10.1002/hec.729.
- [10] J. F. P. Bridges et al., "Conjoint Analysis Applications in Health—a Checklist: A Report of the ISPOR Good Research Practices for Conjoint Analysis Task Force," *Value Heal.*, vol. 14, no. 4, pp. 403–413, Jun. 2011, doi: 10.1016/j.jval.2010.11.013.

- [11] T. A. Domencich and D. McFadden, “Urban travel demand—a behavioral analysis,” 1975.
- [12] George W Torrance, “Measurement of Health State Utilities for Economic Appraisal,” *J. Health Econ.*, vol. 5, 1986.
- [13] K. Zwerina, J. Huber, and W. F. Kuhfeld, “A general method for constructing efficient choice designs,” *Durham, NC Fuqua Sch. Business, Duke Univ.*, 1996.
- [14] W. F. Kuhfeld, “Experimental Design: Efficiency, Coding, and Choice Designs.” Accessed: Aug. 21, 2020. [Online]. Available: <http://support.sas.com/techsup/technote/mr2010c.sas>.
- [15] J. Huber and K. Zwerina, “The Importance of Utility Balance in Efficient Choice Designs,” *J. Mark. Res.*, vol. 33, no. 3, pp. 307–317, Aug. 1996, doi: 10.1177/002224379603300305.
- [16] M. Crabbe, D. Akinc, and M. Vandebroek, “Fast algorithms to generate individualized designs for the mixed logit choice model,” *Transp. Res. Part B Methodol.*, vol. 60, pp. 1–15, 2014, doi: 10.1016/j.trb.2013.11.008.
- [17] H. Aizaki and K. Nishimura, “Design and Analysis of Choice Experiments Using R: A Brief Introduction,” *Agric. Inf. Res.*, vol. 17, no. 2, pp. 86–94, 2008, doi: 10.3173/air.17.86.
- [18] A. B. Hauber et al., “Statistical Methods for the Analysis of Discrete Choice Experiments: A Report of the ISPOR Conjoint Analysis Good Research Practices Task Force,” *Value Heal.*, vol. 19, no. 4, pp. 300–315, 2016, doi: 10.1016/j.jval.2016.04.004.
- [19] K. E. Train, *Discrete choice methods with simulation, second edition, vol. 9780521766555*. Cambridge University Press, 2009.
- [20] J. Oliva-Moreno, L. M. Peña-Longobardo, L. García-Mochón, M. Del Río Lozano, I. M. Metcalfe, and M. Del Mar García-Calvente, “The economic value of time of informal care and its determinants (The CUIDARSE Study),” *PLoS One*, vol. 14, no. 5, May 2019, doi: 10.1371/journal.pone.0217016.