Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

Efficient and Accessible Experimental Designs for Discrete Choice Experiments: DCEtool

Daniel Pérez Troncoso
Universidad de Granada

Abstract

In recent years, computational methods have been employed for the construction of efficient discrete choice experiments. However, many of these methods are only available on paid software or free software without a user-friendly interface. This paper introduces DCEtool, a free app coded in **shiny** R to make good design properties accessible to new and experienced researchers.

Keywords: discrete choice experiments, conjoint analysis, **shiny** R.

1. Introduction: Efficient Experimental Designs for DCEs

For the last two decades, Discrete Choice Experiments (DCE) design techniques have been the subject of an intense debate. Traditionally, researchers have relied on orthogonality to reduce full factorial designs (Johnson, Lancsar, Marshall, Kilambi, Muhlbacher, Regier, Bresnahan, Kanninen, and Bridges 2013). However, many soon pointed out that orthogonality was not appropriate due to the non-linearity of discrete choice models (Bliemer and Rose 2010).

To avoid problems related to non-linearity, many researchers started using statistically efficient designs. Statistically efficient designs seek to increase the precision of the estimates by reducing the design D-error (Rose and Bliemer 2009). The reduction of the D-error implies, in short, the minimization of the standard errors of the parameters estimated from the DCE data. Due to the iterative procedure required to obtain efficient designs, a software-based optimization routine needs to be used. Some examples are Ngene (ChoiceMetrics 2018), the **choiceff** macro for SAS (Kuhfeld 2007), the **dcreate** module for Stata (StataCorp LLC 2017), or the **idefix** package for R (R Core Team 2021). However, to date there were no free software alternatives with a user-friendly interface.

This article aims to introduce DCEtool, an R package with a visual interface coded in **Shiny** that makes good design properties accessible to both new and experienced researchers. DCEtool can create and decode a DCE design matrix, construct a local interactive survey from it, and analyze the responses using discrete choice models. DCEtool incorporates a) code from the **idefix** package to create the experimental design and present the interactive survey, b) code from the **survival** and **mlogit** packages to analyze the results, and c) new code for the visual interface and data management.

Since DCEtool includes all the resources necessary to conduct and analyze DCEs, the researcher can create a survey, answer it, and estimate a discrete choice model in a matter of minutes. Thus, the pedagogical potential of DCEtool can be of great value to new researchers. In addition, this software includes a built-in functionality to implement the proposal of Bliemer and Rose (Bliemer and Rose 2010), serial discrete choice experiments. Since serial designs can be very time-consuming to implement manually, this app offers the possibility to implement them at no additional cost.

2. Requirements and installation

2.1. Requirements

The program has been tested in most platforms: Windows, macOS, and Linux (Ubuntu and POP_OS!). For it to work, the package requires both R (R Core Team 2021) and RStudio (RStudio Team 2021) (tested in versions 4.1.0 and 1.4.1106 respectively). The other required packages will be installed during the DCEtool installation. There are no specific hardware requirements, but program performance will be worse in computers with less performant CPUs and lower and slower RAMs (go to section 4.2 for some benchmarks).

2.2. Installation

DCEtool is available on CRAN so its installation is similar to any other package from the official repository:

```
> install.packages("DCEtool")
```

Once installed, the graphical interface will appear after typing the following commands:

```
> library(DCEtool)
> DCEtool()
```

However, the most recent iteration of the software is likely to be found earlier on GitHub (github.com/danielpereztr/DCEtool) than on CRAN. It can be directly installed in RStudio using the following commands:

```
> install.packages("devtools")
> library(devtools)
> install_github("danielpereztr/DCEtool")
> library(DCEtool)
> DCEtool()
```

Attributes	Levels
Effectiveness	70%
	80%
	90%
Required dosage	1 dose
	2 doses
Adverse events	1 in 1000 patients
	1 in 500 patients
	1 in 100 patients
Out-of-pocket cost	100€
	150€
	200€

Table 1: Selection of attributes and levels used to illustrate the procedure of constructing a statistically efficient design with DCEtool

Due to CRAN policy, CRAN packages cannot be updated more than one time every 1 or 2 months, so, during specific periods, the code hosted at GitHub might correspond to a newer version. If the above lines of code do not work on Linux, go to section 4.1.

3. Instructions for use

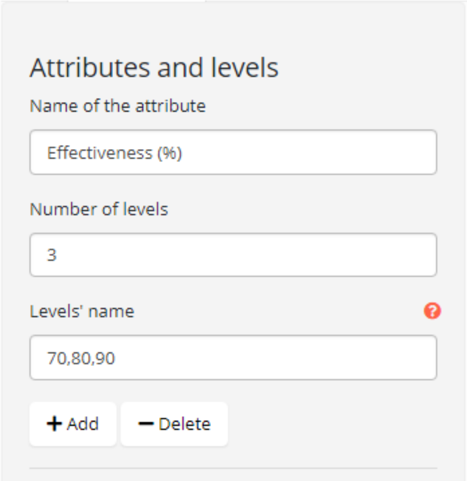
3.1. Attributes, levels, choice sets, alternatives, and prior parameters

On the first page of the virtual environment, the user can press on "Create a DCE" to move to the "Parameters" tab. In "Parameters", the user will be asked to input all the settings required to construct the experimental design.

The first step is to carry out a selection of attributes and levels. In this case, the set of attributes and levels in Table 1 will be used as an example. As we can observe, there are 4 attributes with 3, 2, 3, and 3 levels. A full factorial design from the selection in Table 1 would lead to $3^3 \times 2 = 54$ alternatives. If these alternatives were combined into pairs, the total number of resulting choice sets would be $(54 \times 53)/2 = 1431$, a number of sets too large to be used in a survey. Since 70% of DCEs use between 3 and 7 attributes (Marshall, Bridges, Hauber, Cameron, Donnalley, Fyie, and Johnson 2012), most designs must be reduced from the full factorial design, i.e., choosing a subset of choice sets selected among all the choice sets in the full factorial design.

The selection of an optimal subset of choice sets can be performed through the coordinate exchange algorithm (CEA) from the **idefix** package that has been incorporated into DCEtool. For the software to yield a statistically efficient design using the CEA, the user needs to input all the required parameters in the left panel of the "Parameters" tab. First, each attribute must be introduced individually, indicating how many levels it has and labeling each level. The names of the levels must be introduced in the same line and separated by commas (see example in Figure 1). The table will be updated each time the user clicks "Add +". The "Load example data" button loads the selection of attributes and levels in Table 1.

Once the attributes and levels have been specified, the researcher must decide the number



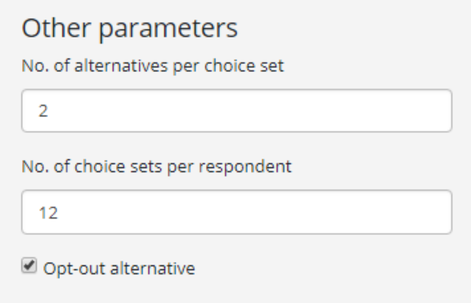
Attributes and levels

Name of the attribute: Effectiveness (%)

Number of levels: 3

Levels' name: 70,80,90

Figure 1: Attribute and levels input in DCEtool



Other parameters

No. of alternatives per choice set: 2

No. of choice sets per respondent: 12

☑ Opt-out alternative

Figure 2: Choice sets per survey and alternatives per choice set

of alternatives per choice set and choice sets per respondent. The program also allows the inclusion of an opt-out option by clicking on the check box. To illustrate this example, we will use 12 choice sets per respondent and two alternatives per choice set plus an opt-out alternative (Figure 2).

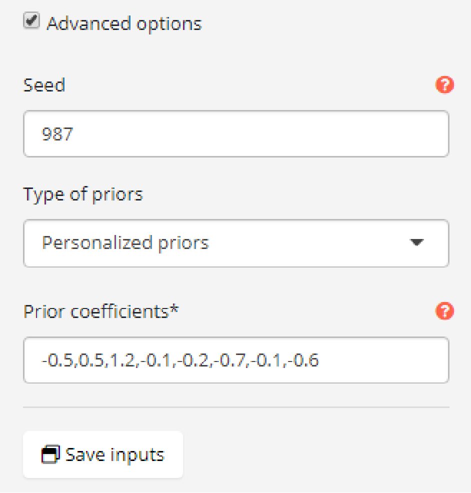
The user can also specify several advanced options. First, a “seed number” can be used if the user wants to generate always the same design under the same parameters. Since randomness plays an essential role in the optimization algorithm, it is recommended to use a seed number (any number that the user can remember).

Then, the user can decide whether to use zero or non-zero priors. The priors are a prediction of the coefficients of the model to be estimated. Usually, the priors are obtained from pilot experiments. If no priors are available, or the user is creating a pilot experiment, the drop-down list should specify "Zero (pilot design)". However, if the user knows some prior values, the drop-down list can be changed to "Personalized priors".

When "Personalized priors" are selected, the textbox below needs to contain the prior parameters in order and separated by commas. To introduce the prior parameters, the user can follow the scheme in Table 2. As we can see, there are eight parameters: a total of eleven levels minus the first level of each attribute (which is constricted to zero due to dummy coding) plus one parameter due to the opt-out alternative (11-4+1). If there were no opt-out

$\tilde{\beta}_{opt-out}$	$\tilde{\beta}_{effect.80}$	$\tilde{\beta}_{effect.90}$	$\tilde{\beta}_{doses.2}$	$\tilde{\beta}_{adv.1in500}$	$\tilde{\beta}_{adv.1in100}$	$\tilde{\beta}_{cost.150}$	$\tilde{\beta}_{cost.200}$
-0.5	0.5	1.2	-0.1	-0.2	-0.7	-0.1	-0.6

Table 2: Prior coefficients (example) to be inputted in DCEtool for the selection of attribute levels in Table 1.


☑ Advanced options

Seed

987

Type of priors

Personalized priors

Prior coefficients*

-0.5,0.5,1.2,-0.1,-0.2,-0.7,-0.1,-0.6

Figure 3: Advanced options. Random seed number and personalized priors.

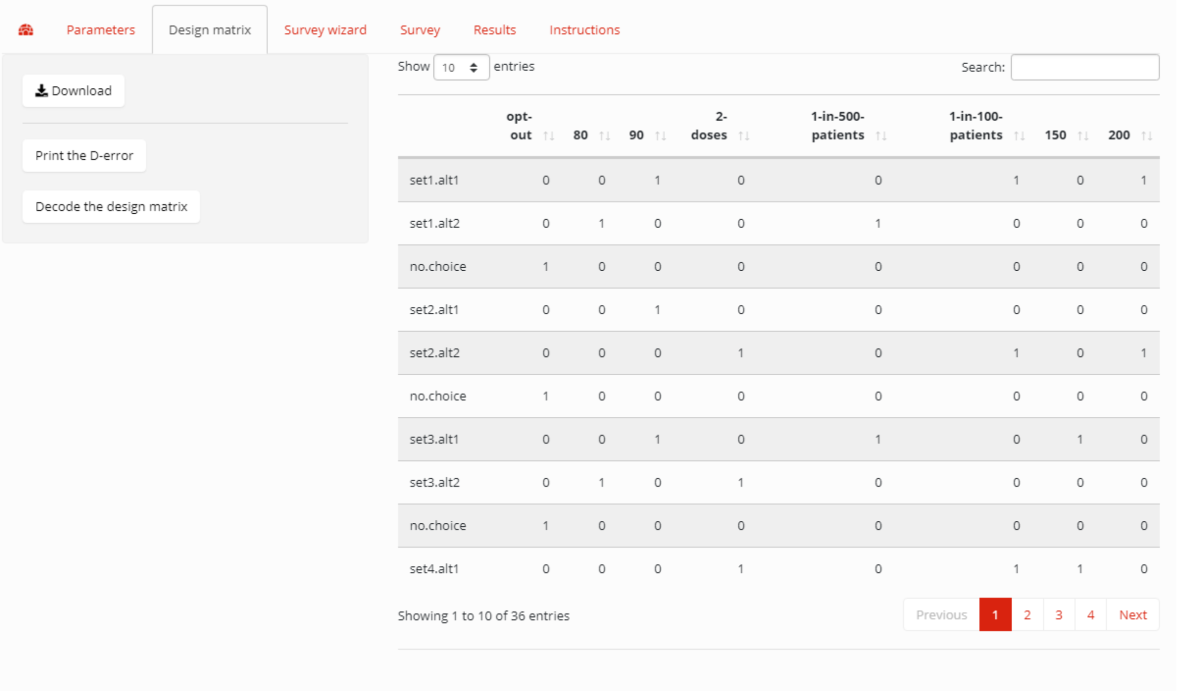
alternative, there would be seven parameters. Thus, the prior coefficients need to be written in the same order that we specified in Table 1, reserving the first place for the coefficient of the opt-out alternative (if any) (Figure 3). Finally, the "Save inputs" button must be clicked to save the parameters entered.

3.2. Design matrix

Once the inputs have been saved, an efficient experimental design will be computed if the user moves to the "Design matrix" tab. The calculation time of the experimental design may vary depending on the characteristics of the CPU and RAM used by the computer. After waiting for several seconds or minutes (depending on the size of the experimental design), when the loading animation is finished, the design matrix should appear in the form of a table (Figure 4). At this point, the user could download and use the experimental design generated in a paper survey.

The design matrix allocates the attribute levels to each choice set in the survey (Rose and Bliemer 2014). This matrix can be saved as an Excel file by clicking on the "Download" button. The user could obtain the layout of the survey by interpreting the resulting Excel. Table 3 can be used to decode the position of the attributes and levels in each alternative of each choice set. Each row represents a different alternative. For example, the first alternative of the first choice set has 90% effectiveness, one dose, adverse events in 1 in 1000 patients, and a cost of 100€. If all the levels from an attribute in the table are zero, that alternative contains the omitted level (for example, one dose in the case of the second attribute).

However, the user does not need to decode the design matrix. By pressing on “Decode the design matrix”, a text output will indicate the composition of the choice sets in the discrete

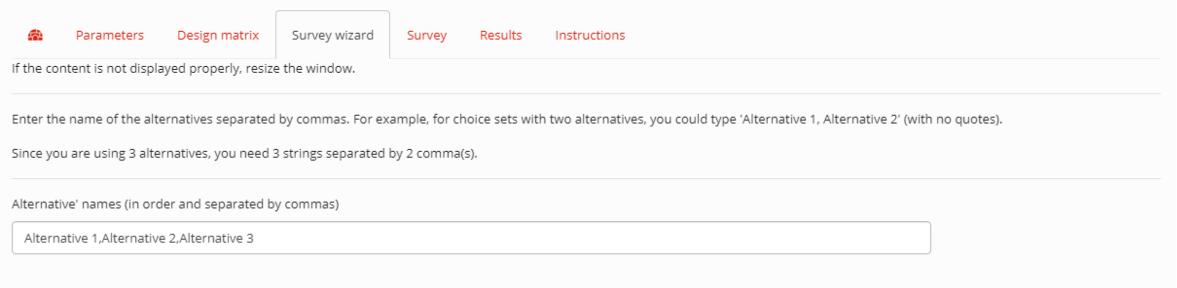


	opt-out	80	90	2-doses	1-in-500-patients	1-in-100-patients	150	200
set1.alt1	0	0	1	0	0	1	0	1
set1.alt2	0	1	0	0	1	0	0	0
no.choice	1	0	0	0	0	0	0	0
set2.alt1	0	0	1	0	0	0	0	0
set2.alt2	0	0	0	1	0	1	0	1
no.choice	1	0	0	0	0	0	0	0
set3.alt1	0	0	1	0	1	0	1	0
set3.alt2	0	1	0	1	0	0	0	0
no.choice	1	0	0	0	0	0	0	0
set4.alt1	0	0	0	1	0	1	1	0

Figure 4: Design matrix.

	$\tilde{\beta}_{opt-out}$	$\tilde{\beta}_{effect.80}$	$\tilde{\beta}_{effect.90}$	$\tilde{\beta}_{doses.2}$	$\tilde{\beta}_{adv.1in500}$	$\tilde{\beta}_{adv.1in100}$	$\tilde{\beta}_{cost.150}$	$\tilde{\beta}_{cost.200}$
	↑	↑	↑	↑	↑	↑	↑	
V1	alt3.cte	Var12	Var13	Var22	Var32	Var33	Var42	Var43
sel1.alt1	0	0	1	0	0	0	0	0
sel1.alt2	0	0	0	1	0	1	0	1
sel1.alt2	1	0	0	0	0	0	0	0
				...				

Table 3: Design matrix correspondence with model coefficients



If the content is not displayed properly, resize the window.

Enter the name of the alternatives separated by commas. For example, for choice sets with two alternatives, you could type 'Alternative 1, Alternative 2' (with no quotes).

Since you are using 3 alternatives, you need 3 strings separated by 2 comma(s).

Alternative' names (in order and separated by commas)

Alternative 1,Alternative 2,Alternative 3

Figure 5: Alternatives' names in the local survey.

	Code	Result
Large heading	# Text	**Text**
Medium heading	## Text	**Text**
Small heading	### Text	**Text**
Small heading	### Text	**Text**
Bold	**Text**	**Text**
Italics	*Text*	*Text*
Link	[Text](https://git.io/cnT_vA)	Text

Table 4: Text formatting in Markdown. For further details, visit the link inserted in the table.

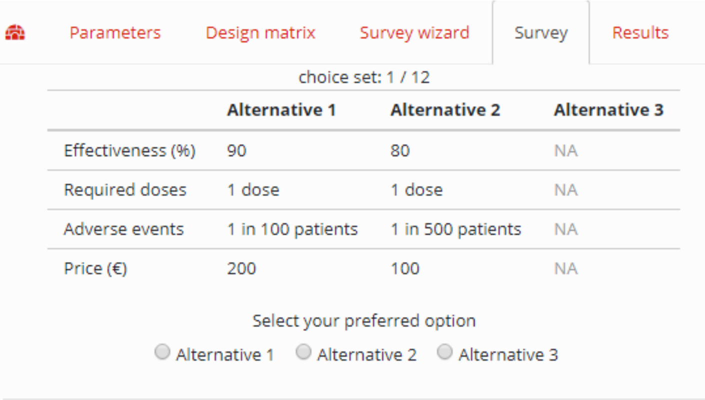
choice experiment. In addition, the D(B)-error of the design can be obtained if the user needs to report it in the memory of the research.

3.3. Local survey

The "Survey wizard" tab provides an interface to set up a local survey in the app. In the first step, the user needs to introduce the alternatives' names. For example, if we want to call the alternatives "Alternative A", "Alternative B", and "Alternative C", we need to introduce the names separated by commas in the textbox (Figure 5)

The next step is the creation of an introductory and end text. Both texts will be displayed repeatedly at the beginning and end of each survey so that the user will be able to explain to respondents how to proceed. The texts must be written in Markdown language, a flexible language that allows the user to introduce typographical variations, tables and links to explain the content of the survey. The user can use Table 4 as a guide for writing the introductory and end text in Markdown.

On the last page of the "Survey wizard", the user can decide whether to use the Bliemer and Rose (2010) approach: serial DCEs. In serial DCEs, when a new response is recorded, a new discrete choice model is estimated from the available data (pooled from all responses collected so far). Then, if the estimates are significant, they are used as prior coefficients for the next survey. The discrete choice model used is a conditional logit model (CLM) (McFadden 1974) estimated with the function "clogit" from the **survival** R package. When the "use serial approach" checkbox is clicked, the next page allows the user to input a number, after which the serial approach will start. If the serial approach begins after several responses, the user can avoid endogeneity bias (Rose 2012).



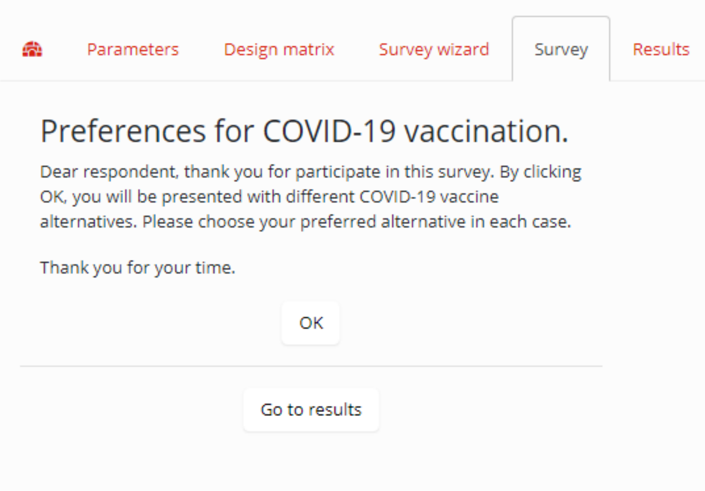
choice set: 1 / 12

	Alternative 1	Alternative 2	Alternative 3
Effectiveness (%)	90	80	NA
Required doses	1 dose	1 dose	NA
Adverse events	1 in 100 patients	1 in 500 patients	NA
Price (€)	200	100	NA

Select your preferred option

○ Alternative 1 ○ Alternative 2 ○ Alternative 3

Figure 6: Choice set presented in the local survey.



Preferences for COVID-19 vaccination.

Dear respondent, thank you for participate in this survey. By clicking OK, you will be presented with different COVID-19 vaccine alternatives. Please choose your preferred alternative in each case.

Thank you for your time.

Figure 7: Introductory text.

3.4. Survey

Once the "Create the survey" button has been pushed in the "Survey wizard" tab, the user can move to "Survey". In the "Survey" tab, the user can respond directly to the discrete choice experiment just created (Figure 6). This tab is based on the survey functionality coded in the **idefix** package. Here, the code in the **idefix** package has been substantially modified to show the Markdown introductory and end text (Figure 7), load a new survey when a new response is recorded, and incorporate the serial strategy by Bliemer and Rose.

Thus, each time a response is collected (when all the choice sets from the survey are answered), a new survey will appear. Since the introductory text and the end text are shown at the beginning and the end of the DCE, the "Survey" tab can be used in an actual discrete choice experiment. While the final text is displayed, the "next respondent" and "go to results" buttons will also be displayed. If the user clicks in the "Next respondent" button while using a serial approach, the following survey may take a few seconds to appear (see Section 4.2). By clicking on "Go to results", the user will be redirected to the "Results" tab. The user should not switch tabs before the survey has finished, or response data could be lost.

	Attribute levels				Choice	Choice set id	Alt. id	Respondent id
					↑	↑	↑	↑
	null	80	...	150	bin.responses	gid	alt	pid
set1.alt1	0	0	...	0	1	1	1	1
set1.alt2	0	1	...	1	0	1	2	1
set1.alt3	1	0	...	0	0	1	3	1

Table 5: Interpretation of the result table. The red row denotes the alternative chosen according to the "Choice" column.

3.5. Results

In the "Results" tab, a table including the design matrix and the responses will be displayed. By clicking on "Download", the user can save an Excel file summarizing all the information gathered in the DCE survey. The table can be interpreted as shown in Table 5.

From the results presented in Table 5, a CLM and a mixed (random parameters) logit model (XLM) can be estimated by clicking on the corresponding buttons. However, notice that the XLM requires a large sample and will not work if the sample is small[1].

When the user clicks on the model button, the estimation results will be displayed as follows:

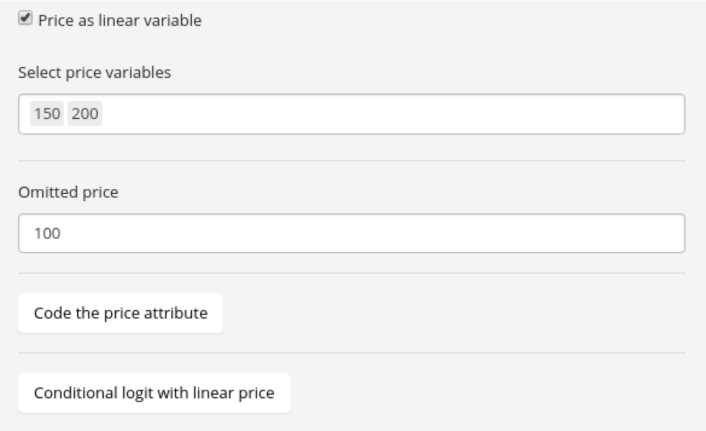
```
Call:
coxph(formula = Surv(rep(1, 144L), bin.responses) ~ `opt out` +
    `80` + `90` + `2 doses` + `1 in 500 patients` + `1 in 100 patients` +
    `150` + `200`, data = results, method = "exact")

  n= 144, number of events= 48

                         coef  exp(coef)   se(coef)      z Pr(>|z|)
`opt out`            0.465556   1.592899   1.046577  0.445 0.656438
`80`                 4.323347  75.440672   1.123857  3.847 0.000120 ***
`90`                 4.630731 102.589075   1.123249  4.123 3.75e-05 ***
`2 doses`           -0.925003   0.396530   0.767844 -1.205 0.228329
`1 in 500 patients`  0.536390   1.709823   0.872089  0.615 0.538513
`1 in 100 patients` -0.186846   0.829571   0.809779 -0.231 0.817519
`150`               -2.260640   0.104284   0.846949 -2.669 0.007604 **
`200`               -4.978125   0.006887   1.320663 -3.769 0.000164 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

                    exp(coef) exp(-coef) lower .95 upper .95
`opt out`           1.593e+00  6.278e-01 2.048e-01  12.38900
`80`                7.544e+01  1.326e-02 8.336e+00 682.70686
`90`                1.026e+02  9.748e-03 1.135e+01 927.28320
`2 doses`           3.965e-01  2.522e+00 8.804e-02   1.78594
`1 in 500 patients` 1.710e+00  5.849e-01 3.095e-01   9.44657
```

[1] The user is recommended to save the results before performing any estimation.



☑ Price as linear variable

Select price variables

150 200

Omitted price

100

Figure 8: Cost levels as a single coefficient

```
`1 in 100 patients` 8.296e-01  1.205e+00 1.697e-01   4.05638
`150`               1.043e-01  9.589e+00 1.983e-02   0.54845
`200`               6.887e-03  1.452e+02 5.175e-04   0.09166

Concordance= 0.866  (se = 0.03 )
Likelihood ratio test= 70.63  on 8 df,   p=4e-12
Wald test            = 33.23  on 8 df,   p=6e-05
Score (logrank) test = 57.27  on 8 df,   p=2e-09
```

The output will provide two tables, one for the coefficients and the other for the odds ratios. From the coefficients' table, the user can observe the amount of utility that the respondent obtains from each level. To estimate the willingness-to-pay (WTP), the user can code the cost levels as a single linear level. It can be easily done by clicking on "Price as a liner variable", indicating the cost levels in the table (150 and 200 in this case), and introducing the omitted cost level (100) (Figure 8). Then, the user can click on "Code the price attribute" to add the new variable to the table and "Conditional logit with linear price" to obtain the following output:

```
Call:
coxph(formula = Surv(rep(1, 144L), bin.responses) ~ `opt out` +
    `80` + `90` + `2 doses` + `1 in 500 patients` + `1 in 100 patients` +
    pricevars, data = results, method = "exact")

  n= 144, number of events= 48

                         coef exp(coef)  se(coef)      z Pr(>|z|)
`opt out`            -4.45494   0.01162   1.59660 -2.790  0.00527 **
`80`                  4.42305  83.34999   1.09410  4.043 5.29e-05 ***
`90`                  4.71279 111.36208   1.10700  4.257 2.07e-05 ***
`2 doses`            -0.95658   0.38421   0.76386 -1.252  0.21046
`1 in 500 patients`   0.52786   1.69530   0.87713  0.602  0.54731
`1 in 100 patients`  -0.22974   0.79474   0.79793 -0.288  0.77341
```

```
pricevars           -0.04876   0.95241   0.01246 -3.914 9.08e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

                    exp(coef) exp(-coef) lower .95 upper .95
`opt out`             0.01162   86.05076 5.084e-04    0.2656
`80`                 83.34999    0.01200 9.763e+00  711.5512
`90`                111.36208    0.00898 1.272e+01  975.0221
`2 doses`             0.38421    2.60277 8.597e-02    1.7170
`1 in 500 patients`   1.69530    0.58987 3.038e-01    9.4593
`1 in 100 patients`   0.79474    1.25827 1.664e-01    3.7969
pricevars             0.95241    1.04996 9.294e-01    0.9760

Concordance= 0.866  (se = 0.03 )
Likelihood ratio test= 70.54  on 7 df,   p=1e-12
Wald test            = 32.09  on 7 df,   p=4e-05
Score (logrank) test = 57.25  on 7 df,   p=5e-10
```

As reflected in the model summary, the coefficients "150" and "200" have been replaced by "pricevars". This new variable indicates the cost of each alternative with an integer number in each row (see the last variable in the table in the "Results" tab).

From the model summary, the user can estimate the WTP of each level using the following formula:

$$WTP_{effect.90} = -\tilde{\beta}_{effect.90}/\tilde{\beta}_{pricevars} = -4.71279/-0.04876 = 96,65$$

Therefore, the average respondent is willing to pay up to 96.65€ for a 90% of effectiveness.

4. Code and details

4.1. Installation on Linux systems

DCEtool can be easily installed in systems running Linux using the install.packages(“DCEtool”) command. However, if the user wants to install **DCEtool** from GitHub, the **devtools** package must be installed. Since it is known that users have problems installing the **devtools** package in machines running Linux, this section aims to provide the steps required to install **devtools** and **DCEtool** from GitHub.

Since the installation of **devtools** requires superuser permissions, the following code has to be introduced in the Terminal:

```
\$ sudo apt-get -y install libcurl4-gnutls-dev libxml2-dev libssl-dev
\$ sudo su
\$ R
> install.packages('devtools', repos='http://cran.rstudio.com/')
```

If the procedure finishes successfully, the installation of **DCEtool** must be available by typing:

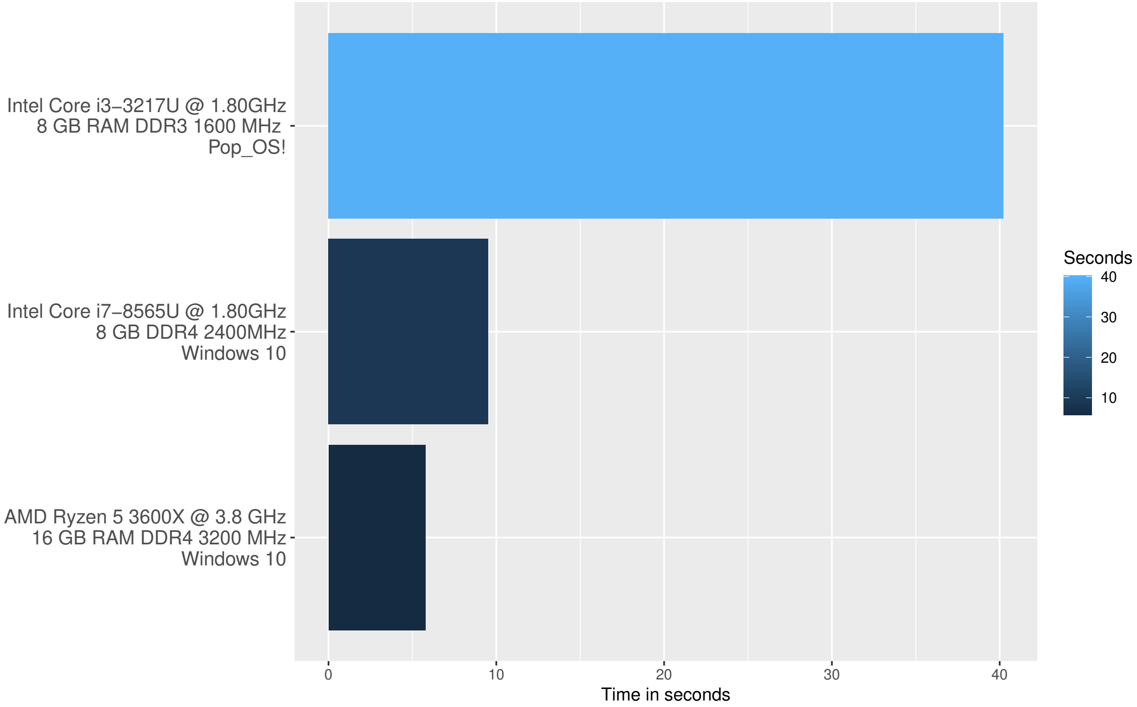


Figure 9: Time for computing a design with the selection in Table 1, 8 choice sets, and 2 alternatives per choice set.

```
> library(devtools)
> install_github('danielpereztr/DCEtool')
```

Note that the **APT** library must be installed in order to run the previous code. This library is usually installed by default in Debian, Ubuntu, and similar distros as Linux Mint and Pop-OS!.

4.2. Computational time

As has been mentioned before, the computational times of the efficient design depend on the computer's CPU and RAM. It is therefore important to not only consider computational time when creating the starting design but also when using a serial efficient design. A benchmark test in three different systems lead us to the results in Figure 9. Therefore, if the user wants to employ the serial strategy, it is highly recommended to use a computer with a high-capacity CPU to reduce respondent-to-respondent loading times.

5. Future development

Today, DCEtool is a local tool with clear practical and pedagogical applications. However, the value of the application could be increased with the inclusion of the following functionalities:

- Estimation of the willingness-to-pay after coding the price attribute.

- A new tab to allow the user to upload and manage their own data.
- An online implementation of the survey.

The most complex task is the online implementation of the survey. Any contribution in this regard can be made through https://github.com/danielpereztr/DCEtool/ and will be highly appreciated.

References

Bliemer MC, Rose JM (2010). *Serial choice conjoint analysis for estimating discrete choice models.* Emerald Group Publishing Limited. doi:10.1108/9781849507738-006.

ChoiceMetrics (2018). "1.2 User Manual & Reference Guide." *ChoiceMetrics Pty Ltd.: Sydney, Australia.* URL http://www.choice-metrics.com/NgeneManual120.pdf.

Johnson FR, Lancsar E, Marshall D, Kilambi V, Muhlbacher A, Regier DA, Bresnahan BW, Kanninen B, Bridges JF (2013). "Constructing experimental designs for discrete-choice experiments: report of the ISPOR conjoint analysis experimental design good research practices task force." *Value in health*, **16**(1), 3–13. doi:10.1016/j.jval.2012.08.2223.

Kuhfeld WF (2007). "Marketing research methods in SAS: experimental design, choice, conjoint, and graphical techniques." Citeseer. URL http://support.sas.com/resources/papers/tnote/tnote_marketresearch.html#market.

Marshall D, Bridges JFP, Hauber B, Cameron R, Donnalley L, Fyie K, Johnson FR (2012). "Conjoint Analysis Applications in Health — How are Studies being Designed and Reported?" *The Patient: Patient-Centered Outcomes Research 2010 3:4*, **3**(4), 249–256. ISSN 1178-1661. doi:10.2165/11539650-000000000-00000. URL https://link.springer.com/article/10.2165/11539650-000000000-00000.

McFadden D (1974). "Conditional logit analysis of qualitative choice behavior." In *Frontiers in Econometrics.*, pp. 105–42. Academic Press, New York, NY. URL http://elsa.berkeley.edu/reprints/mcfadden/zarembka.pdf.

R Core Team (2021). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rose J (2012). "Workshop on experimental designs. Danish Choice Modelling Day 2012 Experimental Designs."

Rose JM, Bliemer MC (2009). "Constructing efficient stated choice experimental designs." *Transport Reviews*, **29**(5), 587–617. doi:10.1080/01441640902827623.

Rose JM, Bliemer MC (2014). "Stated choice experimental design theory: The who, the what and the why." In *Handbook of Choice Modelling*, pp. 152–177. ISBN 9781781003152. doi:10.4337/9781781003152.00013.

RStudio Team (2021). "RStudio: Integrated Development Environment for R." URL http://www.rstudio.com/.

StataCorp LLC (2017). “Stata Statistical Software.” URL https://www.stata.com/.

Affiliation:

Daniel Pérez-Troncoso
Department of Applied Economics
Faculty of Economics and Business
Universidad de Granada
Paseo de Cartuja, 7
18011 Granada
E-mail: danielperez@ugr.es
URL: http://danielpereztr.es/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd