An R Markdown program to create the experimental design for a Discrete Choice Experiment (DCE) exploring online help seeking in socially anxious young people

Complete Survey Design Program

Matthew P Hamilton[1,*]

26 October 2022

[1] Orygen, Parkville, Australia

[*] Correspondence: Matthew P Hamilton <matthew.hamilton@orygen.org.au>

Suggested citation: "Matthew Hamilton (2022). dce_sa_design: An R Markdown program to create the experimental design for a Discrete Choice Experiment (DCE) exploring online help seeking in socially anxious young people. Zenodo. https://doi.org/10.5281/zenodo.6626256"

1 About this code

1.1 Motivation

This program was used to develop a pilot survey design, analyse pilot survey responses and generate a final survey choice efficient design and choice cards for for a Discrete Choice Experiment study that is currently being written up. Future versions of this program will include details of the parent study.

1.2 Status

This code has been adapted from the code originally used in the study to make it easier to generalise. If you have access to the study dataset, this code will still generate an identical design to that used in our study. Future releases of this program will include synthetic data to allow those without access to the study dataset to run it from start to finish. The replication code will produce an efficient design with different choice sets to that produced in the original study.

1.3 Use

Note, for reporting purposes this program can be executed in full so long as you have access to the external datasets referenced in the code. These datasets will only be available on completion of a study. When using the program to help design the study it is necessary to only execute the first part (design of pilot study), then collect pilot data, before returning to execute the remaining parts of the program. It should also be noted that some of the steps in this program involve interactivity - they generate a prompt that a user must respond to before proceeding. Therefore, **this code should be run step by step** (i.e run one chunk at a time and do not try to run the program by knitting the R Markdown version of this code). Although it would be possible to add work-arounds to the interactivity issue, running the program by knitting the RMD version is still not recommended as it will prevent the documents generated by this program from rendering properly.

2 Prepare workspace

2.1 Install and load required libraries

If you do not already have the required libraries to run this program installed, you can do so by un-commenting and running the following lines.

```
# devtools::install_github("ready4-dev/ready4")
# devtools::install_github("ready4-dev/mychoice") # add lwgeom to imports
# devtools::install_github("ready4-dev/ready4use")
```

Next we load the libraries required to run this program.

```
library(ready4)
library(ready4use)
library(mychoice)
```

2.2 Specify whether program is a reproduction or replication

We begin by declaring whether this program is to be executed in order to reproduce the design of the original study or to replicate that study (which will produce a survey design with similar but different features). If choosing a replication, change the below setting to `F`.

```
reproduce_1L_lgl <- T
```

2.3 Specify data directories

We next specify where our input data can be located and where we wish to write our outputs to. You must supply these details or the rest of this code will not work.

```
paths_ls <- list(input_data_dir_1L_chr = "PROVIDE DETAILS HERE",
                 output_data_dir_1L_chr = "PROVIDE DETAILS HERE",
                 raw_data_fl_nms_chr = "PROVIDE DETAILS HERE")
```

We also specify the online data repository. Note, if you do not have write permissions to the below repository any subsequent command to “share” outputs will not execute correctly.

```
X <- Ready4useRepos(dv_nm_1L_chr = "springtolife",
                    dv_ds_nm_1L_chr = "https://doi.org/10.7910/DVN/VGPIPS",
                    dv_server_1L_chr = "dataverse.harvard.edu")
```

2.4 Reproducibility

We now set a seed to aid reproducibility.

```
set.seed(1001)
```

Having set the seed, it is now likely that if you run the syntax described in this document on your own installation of R you will get identical results to those reported in this document. However, if you do not, it may be that you have a different version of R, or of some of the packages that we used to produce this analysis. We therefore save a record of the software that we have on the machine used for this analysis so this can be made available for comparisons.

```
session_ls <- sessionInfo()
```

3 Specify survey features and priors

Our first main step is to define the key features of the survey (attributes, levels, choice cards, blocks) and our prior expectation of the relative importance of attributes and levels.

3.1 Specify choice attributes and levels

We can now define the attributes and levels for the choices to be included in the survey.

We begin by specifying the following features of our survey:

- the labels used for each choice card alternative;
- the attributes and levels used to describe each alternative;
- the variable names we will use when converting factor attribute levels to dummy variables (reference levels do not need converting);
- the short names we will use when plotting factor attribute levels;
- which of the alternatives is a cost attribute, its currency and frequency of payment; and
- which of the alternatives is an opt-out (no -choice) option.


```
dce_design_ls <- add_design_spec(alternatives_chr = c("Social Anxiety App A", "Social Anxiety App B", "Do not use a social anxiety app"),
     att_lvls_tb = list(c("Outcomes","Information_sharing", "Social","Endorsers", "Cost"),
                        list(c("Provides knowledge and skills to  manage future situations",
                               "Addresses current symptoms",
                               "Addresses current symptoms and provides knowledge and skills to manage future situations"),
                             c("No information is shared with your treating clinician",
                               "Information is shared with your treating clinician in accordance with app policy",
                               "Information is shared with your treating clinician based on settings you control"),
                             c("No discussions with other app users",
                               "Unmoderated discussions with other app users",
                               "Discussions with other app users moderated by trained peers",
                               "Discussions with other app users moderated by mental health clinicians",
                               "Discussions with other app users moderated by both trained peers and mental health clinicians"),
                             c("App has no endorsers",
                               "App is endorsed by respected non experts",
                               "App is endorsed by youth mental health experts"),
```

```
                              c(0,5,15,30,60)),
                         list(c(NA_character_,"curr_sym", "curr_sym_future_sit"),
                              c(NA_character_,"app_policy","user_settings"),
                              c(NA_character_,"unmod_disc", "trained_peer_mod", "clinician_mod", "peer_clinician_mod"),
                              c(NA_character_,"non_expert","expert"),
                              NA_character_),
                         list(c("Future","Current", "Both"),
                              c("None","App policy","User settings"),
                              c("None","Unmoderated", "Peer", "Clinician", "Both"),
                              c("None","Non-expert","Expert"),
                              "Cost")) %>%
       purrr::pmap_dfr(~ tibble::tibble(attribute_chr = ..1,
                                        level_chr = as.character(..2),
                                        continuous_lgl = is.numeric(..2),
                                        dummy_nm_chr = ..3,
                                        distribution_chr = "normal",
                                        short_nms_chr = ..4)),
     cost_att_idx_1L_int = 5L,
     cost_pfx_1L_chr = "$",
     cost_sfx_1L_chr = " per month",
     opt_out_idx_1L_int = 3L,
     session_ls = session_ls)
```

3.2 Create candidate choices matrix

We now create a design matrix of the full factorial of all attribute / level combinations. To do this we create a list specifying the number of attributes for each level and record whether coefficients for each attribute are Continuous (“C”) or Dummy (“D”).

```
dce_design_ls <- add_design_spec(dce_design_ls,
                                 add_cndt_design_mat = T)
```

3.3 Specifiy additional features about our survey

We now provide additional detail about our intended survey design, specifying:

- the total number of choice cards; and

- the number of survey blocks.

```
dce_design_ls <- add_design_spec(dce_design_ls,
                                 nbr_of_sets_1L_int = 30,
                                 nbr_of_blocks_1L_int = 2)
```

3.4 Create matrices of parameter value draws based on prior expectations

We now specify our prior expectation of the values of coefficients for each attribute and an opt out constant. The coefficients supplied in this section were based on study authors' perception of participant feedback at a number of focus groups. We request that 10 values for each parameter be generated and supply a seed value for reproducibility. As this design includes an opt-out alternative, the following command will create two matrices of parameter values: one for the alternative specific constant, the other for the coefficients.

```
dce_design_ls <- add_design_spec(dce_design_ls,
                                 draws_1L_int = 10L,
                                 priors_dbl = c(-0.15, # Opt-out constant
                                                0.5,1, # Outcomes
                                                0.2,0.4, # Information sharing
                                                0.1,0.2,0.3,0.4, # Social
                                                0.1,0.2,# Endorsers
                                                -0.05),
                                 seed_1L_int = 1987L) # Cost
```


4 Create efficient pilot survey design

We can now create the initial efficient design to be used in the pilot survey. Note this step can take a long time (about an hour).

```
if(reproduce_1L_lgl){
  dce_design_ls$efnt_dsn_ls <- append(dce_design_ls$efnt_dsn_ls,
                                     list(ingest(X, fls_to_ingest_chr = c("AAA_pilot_design_ls"), metadata_1L_lgl = F) %>% purrr::pluck("pi
                                          stats::setNames(paste0("Set_", length(dce_design_ls$efnt_dsn_ls) + 1)))
  }else{
    dce_design_ls <- add_design_spec(dce_design_ls,  priors_idx_1L_int = 1L)
}
```

5 Create choice cards for each block

We can now generate HTML choice cards for each block.

```
dce_design_ls <- add_design_spec(dce_design_ls,
                                 add_choice_cards_1L_lgl = T,
                                 transform_att_nms_1L_lgl = T)
```

6 Preview survey

We can now preview all of our choice cards in an interactive Shiny app.

```
launch_survey_preview(dce_design_ls, block_1L_int = 1L)
```

```
launch_survey_preview(dce_design_ls, block_1L_int = 2L)
```


7 Share work

The final step is to share our work with others in an online repository. Note, you will need to have previously supplied values for `X` that correspond to a repository for which you have write permissions to run this part of the code successfully.

```
Y <- share(X,
           obj_to_share_xx = dce_design_ls,
           fl_nm_1L_chr = ifelse(reproduce_1L_lgl,"DDD_pilot_dce_design_ls","CCC_pilot_dce_design_ls"),
           description_1L_chr = paste0("Pilot survey design specification - ",
                                       ifelse(reproduce_1L_lgl,"reproduction","replication")," (output of dce_sa_design program)"))
```

8 Ingest data

We ingest the responses to our pilot survey.

```
pilot_ds_tb <- get_table_from_loc_file(paste0(paths_ls$input_data_dir_1L_chr,
                                              "/",
                                              paths_ls$pilot_ds_fl_nm_1L_chr),
                                       heading_rows_1L_int = 3L,
                                       force_numeric_1L_lgl = T)
```

```
## Rows: 9 Columns: 81
## -- Column specification --------------------------------------------------------------------------------------------------------------------------------
## Delimiter: ","
## chr (81): StartDate, EndDate, Status, IPAddress, Progress, Duration (in seconds), Finished, RecordedDate, ResponseId, RecipientLastName
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```


We also retrieve details about the design of our pilot survey.

```
dce_design_ls <- ingest(X,
                        fls_to_ingest_chr = ifelse(!reproduce_1L_lgl,"CCC_pilot_dce_design_ls","DDD_pilot_dce_design_ls"),
                        metadata_1L_lgl = F)
```

9 Analyse data

The analysis algorithm we are going to apply involves random sampling so a seed must be set to make results reproducible. Note, some random seeds "NULL" values will be generated that will prevent the analysis executing in full. If this occurs, changing the random seed and rerunning the code will address this.

```
dce_design_ls <- add_design_spec(dce_design_ls,
                                 pilot_ds_tb = pilot_ds_tb,
                                 choice_var_pfx_1L_chr = "DCE_B",
                                 constraints_ls = list(Endorsers_non_expert = c(0.005, Inf),
                                                       Endorsers_expert = c(0.006, Inf),
                                                       Cost = c(-Inf, -0.0499)),
```

```
                                draws_1L_int = 100L,
                                seed_1L_int = 1985,
                                set_idx_1L_int = 1L)
```

10 Update survey design

We now update the survey experimental design based on analysis of pilot survey results.

```
if(reproduce_1L_lgl){
  dce_design_ls$efnt_dsn_ls <- append(dce_design_ls$efnt_dsn_ls,
                                     list(readRDS(file = paste0(paths_ls$output_data_dir_1L_chr,"/Archive/Design_Records/no_app_optout_ls.r
                                          stats::setNames(paste0("Set_", length(dce_design_ls$efnt_dsn_ls) + 1)))
  }else{
    dce_design_ls <- add_design_spec(dce_design_ls,  priors_idx_1L_int = 1L, set_idx_1L_int = 1)
}
```

11 Create choice cards for each block

We can now generate HTML choice cards for each block.

```
dce_design_ls <- add_design_spec(dce_design_ls,
                                 add_choice_cards_1L_lgl = T,
                                 block_idxs_ls = {if(reproduce_1L_lgl){
                                   list(as.integer(c(3,5,6,7,8,9,12,14,15,18,19,20,23,24,28)),
                                    as.integer(setdiff(1:dce_design_ls$choice_sets_ls$nbr_of_sets_1L_int,
                                                       c(3,5,6,7,8,9,12,14,15,18,19,20,23,24,28)))) %>%
                                     stats::setNames(paste0(paste0("block_",1:2),"_int"))}else{list()}},
                                 transform_att_nms_1L_lgl = T)
```

12 Preview survey

We can now preview all of our choice cards in an interactive Shiny app.

```
launch_survey_preview(dce_design_ls, block_1L_int = 1L)
```

```
launch_survey_preview(dce_design_ls, block_1L_int = 2L)
```

13 Share work

The final step is to share our work with others in an online repository. Note, you will need to supply your own repository details to run this part of the code successfully as you will not have write permissions to the repository `X` that we specify below.

```
Y <- share(X,
           obj_to_share_xx = dce_design_ls,
           fl_nm_1L_chr = ifelse(reproduce_1L_lgl,"DDD_final_dce_design_ls","CCC_dce_design_ls"),
           description_1L_chr = paste0("DCE design specification for final version of survey - ",
                                       ifelse(reproduce_1L_lgl,"reproduction","replication")," (output of dce_sa_design program)"))
```