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

Final survey design sub-routine (reproduction)

Matthew P Hamilton[1,*]

09 June 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 generate the final survey choice efficient design and choice cards 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 minimally adapted from when it was first applied. It has not been formatted for consistency or ease of use and a number of sections have copied almost unchanged from online examples provided by other authors to demonstrate the third party functions (in particular from the idefix package) used in this program. Future releases of this program aim to adopt a more consistent and integrated approach that should make the program easier to follow and adapt.

1.3 Use

When using this code it is important to note 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 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.

```
# utils::install.packages("idefix")
# utils::install.packages("kableExtra")
# utils::install.packages("knitr")
# utils::install.packages("magick")
# utils::install.packages("magrittr")
# utils::install.packages("shiny")
# utils::install.packages("stringr")
# utils::install.packages("webshot")
# utils::install.packages("xfun")
```

Next load the libraries required to run this program.

```
library(magrittr)
```

3 Create custom functions

Next we create a number of functions that we will use in subsequent parts of this program.

```
export_eff_des <- function(survey_features_ls,
                           parallel = FALSE,
                           output_dir,
                           pilot_analysis = NULL,
                           start_des){
  if(is.null(pilot_analysis))
    par.draws <- survey_features_ls$p.d
  else
    par.draws <- pilot_analysis$sample
  no_app_optout_ls <- idefix::Modfed(cand.set = survey_features_ls$candidate_des_mat,
                                     n.sets = survey_features_ls$n_sets,
                                     n.alts = survey_features_ls$n_alts,
                                     no.choice = survey_features_ls$no_choice_lgl,
                                     alt.cte = survey_features_ls$alt_cte,
                                     parallel = parallel,
                                     par.draws = par.draws,
                                     start.des = start_des)
  ## This section needs generalising before the function can be used for other survey designs.
  dir.create(output_dir)
  dir.create(paste0(output_dir,"/block_1"))
  dir.create(paste0(output_dir,"/block_2"))
  saveRDS(no_app_optout_ls,paste0(output_dir,"/no_app_optout_ls.rds")) # Future dev: add prompt before writing to file
  return(no_app_optout_ls)
}
## Choice Card Functions
make_block_choice_tbs_ls <- function(block_ind,
                                     choices_tb){
  purrr::map(block_ind,
             ~ dplyr::filter(choices_tb, startsWith(Choice, paste0("set",.x,"."))))
```

```r
}
make_choice_card <- function(choice_card_sng_tb){
  formatted_tb <- t(choice_card_sng_tb) %>%
    tibble::as_tibble(rownames = "Attribute") %>%
    dplyr::filter(Attribute != "Choice") %>%
    dplyr::rename(`Social Anxiety App 1` = V1,
                  `Social Anxiety App 2` = V2)
  row_names <- formatted_tb %>% dplyr::pull(Attribute)
  formatted_tb <- formatted_tb %>% dplyr::select(-Attribute)
  formatted_tb <- formatted_tb %>%
    as.data.frame()
  rownames(formatted_tb) <- row_names

  formatted_tb %>%
    knitr::kable(escape = F) %>%
    kableExtra::kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"), full_width = F, position = "left") %>%
    kableExtra::column_spec(1,bold = T, border_right = T) %>%
    kableExtra::column_spec(2:3,
                            color = "black", border_right = T)
}
make_one_block_choice_cards_ls <- function(block_choice_tbs_ls){
  purrr::map(1:length(block_choice_tbs_ls),
             ~ make_choice_card(block_choice_tbs_ls %>% purrr::pluck(.x)))
}
save_one_block_choice_cards <- function(block_choice_cards_ls,
                                        save_path_stub,
                                        output_type = ".png"){

  purrr::walk2(block_choice_cards_ls,
               1:length(block_choice_cards_ls),
               ~ save_choice_card_as(.x, # Future dev: add prompt before writing to file
                                     .y,
                                     save_path_stub = save_path_stub,
                                     output_type = output_type)
  )
}
save_choice_card_as <- function(choice_kab,
                                choice_nbr,
```

```
                                save_path_stub,
                                output_type){
  file_path = paste0(save_path_stub,
                     "/choice_",
                     choice_nbr,
                     output_type)
  if(output_type==".png"){
    kableExtra::as_image(choice_kab,
                         file = file_path)
  }else{
    kableExtra::save_kable(choice_kab, # Future dev: add prompt before writing to file
                           file = file_path,
                           self_contained = F)
  }
}
export_choice_cards <- function(survey_features_ls,
                                no_app_optout_ls,
                                output_dir
                                ){
  survey_ls <- idefix::Decode(des = no_app_optout_ls$design,
                              lvl.names = survey_features_ls$lvl_names,
                              coding = survey_features_ls$c_type,
                              c.lvls = survey_features_ls$con_lvls,
                              alt.cte = survey_features_ls$alt_cte,
                              n.alts = survey_features_ls$n_alts,
                              no.choice = survey_features_ls$no_choice_idx)
  saveRDS(survey_ls,paste0(output_dir, # Future dev: add prompt before writing to file
                           "/survey_ls.rds"))
  choices_tb <- tibble::as_tibble(survey_ls$design,
                                  rownames = "Choice") %>%
    dplyr::rename(Outcomes = V1,
                  `Information sharing` = V2,
                  Social = V3,
                  Endorsers = V4,
                  Cost = V5) %>%
    dplyr::filter(!startsWith(Choice, "no"))
  saveRDS(choices_tb,paste0(output_dir, # Future dev: add prompt before writing to file
                            "/choices_tb.rds"))
```

```
  # Needs generalising
  block_1_ind <- sample(1:survey_features_ls$n_sets,survey_features_ls$n_sets/survey_features_ls$n_blocks) %>% sort()
  block_2_ind <- setdiff(1:survey_features_ls$n_sets,block_1_ind)
  blocks_choice_tbs_ls_ls <- purrr::map(list(block_1_ind,
                                             block_2_ind),
                                        ~ make_block_choice_tbs_ls(.x,choices_tb))
  choice_cards_by_block_ls <- purrr::map(blocks_choice_tbs_ls_ls,
                                         ~ make_one_block_choice_cards_ls(.x))
  purrr::walk2(choice_cards_by_block_ls,
               paste0(output_dir,"/block_",c(1:length(choice_cards_by_block_ls))),
               ~ save_one_block_choice_cards(.x, # Future dev: add prompt before writing to file
                                             .y,
                                             output_type = ".html"))
  list(survey_ls = survey_ls,
       choices_tb = choices_tb,
       blocks_choice_tbs_ls_ls = blocks_choice_tbs_ls_ls,
       choice_cards_by_block_ls = choice_cards_by_block_ls)

}
preview_survey <- function(no_app_optout_ls,
                           survey_features_ls,
                           pilot = T){
  attributes <- names(survey_features_ls$lvl_names) %>% stringr::str_replace_all("_"," ")
  labels <- survey_features_ls$lvl_names %>% unname()
  if(pilot)
    i.text <- "Pilot Survey Preview (All choice cards from all blocks)"
  else
    i.text <- "Final Survey Preview (All choice cards from all blocks)"
  b.text <- "Please choose the alternative you prefer"
  e.text <- "Thanks for taking the survey"
  idefix::SurveyApp(des = no_app_optout_ls$design,
                    n.total = survey_features_ls$n_sets,
                    alts = survey_features_ls$alternatives,
                    atts = attributes,
                    lvl.names = labels,
                    c.lvls = survey_features_ls$con_lvls,
                    coding = survey_features_ls$c_typ,
                    buttons.text = b.text,
```

```
                      intro.text = i.text,
                      end.text = e.text,
                      no.choice = survey_features_ls$no_choice_idx,
                      alt.cte = survey_features_ls$alt_cte)
}
```

```
output_dir <- "final_survey"
final_survey_ls <- export_eff_des(survey_features_ls,
                                  parallel = FALSE,
                                  pilot_analysis = pilot_analysis,
                                  start_des = list(pilot_survey_ls$design),
                                  output_dir = output_dir)
```

4 Create choice cards for each block

We can now generate HTML choice cards for each block.

```
choice_cards_final_ls <- export_choice_cards(no_app_optout_ls = final_survey_ls,
                                             survey_features_ls = survey_features_ls,
                                             output_dir = output_dir)
```

5 Preview survey in Shiny App

```
preview_survey(no_app_optout_ls = final_survey_ls,
               survey_features_ls = survey_features_ls,
               pilot = F)
```