Visual Conjoint Experiments with Simulated Instagram Profiles: A step-by-step Guide for Replication using R.

Gaetano Scaduto

2025-02-14

First of all, install the necessary packages and call the libraries.

```
install.packages("magick")
install.packages("openxlsx")
install.packages("dplyr")
install.packages("showtext", dependencies=T)

library(magick)
library(fs)
library(openxlsx)
library(dplyr)
```

Insert the national contexts in which the experiments will be fielded. The code has been designed to be flexible for cross-country research, but works the same for a single country. Yet, directory must be still organized as if the research was conducted in multiple contexts for the code to be executed properly.

```
contexts = c("IT","CZ","SW","FR")
```

Ad this point you assign the attribute and attribute levels of all the information that **is not conveyed by the profile pictures**. The information conveyed by the profile pictures (gender, age, ethinicity) is instead contained in the filename of the profile pictures themselves (e.g.: "black_female_35". More examples here).

```
job=c("Lawyer","Waiter","Entrepreneur","Teacher","Politician")
issue=c("leftneg","leftpos","rightneg","rightpos")
nostalgia=c("past1","past2","future1","future2")
valence=c("corruption1","corruption2", "honesty1", "honesty2")
food=c("vegan","ethnicrich","meatpoor","meatrich")
animal=c("catpoor","catrich","dogpoor","dogrich")
crowd=c("mixedpeople","whitepeople","mixedelite","whiteelite")
```

Insert your input directory (main_path).[1] To see how the directory must be organized, see an example here. The content and name of the directories could be adjusted to accomodate different research designs. In that case, all subsequent reference in the code to the name of such directories must be changed as well.

```
main_path = "your_path_here"
```

Insert your output directory (*output_wd*) and the path for the Excel sheet containing the textual information for each national context. This directory should contain as many subdirectories as the national contexts are and each subdirectory should be named exactly as the contexts in the vector (e.g., IT, FR).

```
output_wd = "your_output_path_here"
```

[1] Note that if working with Google drive, you need to delete the .ini files in the "propic/round/" subfolders.

Insert the Excel file containing, for every context, the names of the politicians and the pieces that compose the info available in their bios (job, candidacy statement).

An example of how this Excel sheet must be structured is available here:

```
data_name_and_info = read.xlsx("your_path_here")
```

The information in this sheet gets reorganized in a list as follows.

```
name_and_info=list()

for(country in contexts)
{
  data_context = data_name_and_info |>
    filter(context==country)

  list_black_female_names =   trimws(strsplit(data_context$female_black_name, ",")[[1]])
  list_black_male_names =   trimws(strsplit(data_context$male_black_name, ",")[[1]])
  list_white_male_names = trimws(strsplit(data_context$male_white_name, ",")[[1]])
  list_white_female_names = trimws(strsplit(data_context$female_white_name, ",")[[1]])

  list_white_surnames_female = trimws(strsplit(data_context$white_surname_female, ",")[[1]])
  list_white_surnames_male = trimws(strsplit(data_context$white_surname_male, ",")[[1]])
  list_black_surnames_female = trimws(strsplit(data_context$black_surname, ",")[[1]])
  list_black_surnames_male = trimws(strsplit(data_context$black_surname, ",")[[1]])

  name_and_info[[country]] = list(Name=list(White=list(Female=list_white_female_names,
                                                     Male=list_white_male_names
                                                     ),
                                          Black=list(Female=list_black_female_names,
                                                     Male=list_black_male_names
                                                     )
                                          ),
                                  Surname=list(White=list(Male=list_white_surnames_male,
                                                          Female=list_white_surnames_female
                                                          ),
                                             Black=list(Male=list_black_surnames_male,
                                                        Female=list_black_surnames_female
                                                        )
                                             ),
                                  Job=list(Female=list(Lawyer=data_context$female_lawyer,
                                                     Waiter=data_context$female_waiter,
                                                     Entrepreneur=data_context$female_entrepreneur,
                                                     Teacher= data_context$female_teacher,
                                                     Politician =data_context$female_politician
                                                     ),
                                           Male=list(Lawyer=data_context$male_lawyer,
                                                   Waiter=data_context$male_waiter,
                                                   Entrepreneur=data_context$male_entrepreneur,
                                                   Teacher= data_context$male_teacher,
                                                   Politician =data_context$male_politician
                                                   )
                                          ),
                                  Candidacy=list(Female=data_context$female_candidacy,
                                                 Male=data_context$male_candidacy
                                                 )
```

```
                                 )

}
```

We then assign the probabilities of seeing black vs white politician (5% vs 95%).

```
probabilities = c(0.0125,0.0125, 0.0125,0.0125,
                  0.2375,0.2375,0.2375,0.2375)
```

We then load the empty template to generate the profiles. Notice that all the sizes and proportion are based exactly on this specific template. Changes in the Instagram template in January 2025 might prompt different choices from the researchers. The template used in the original experiment is available here.

```
template = image_read(paste0(main_path,"Template/template_empty1.png"))
```

```
for(context in contexts) #Looping over all national contexts
{

npics=15000 #number of pictures to generate. Adjust according to need

for(i in 1:npics)
{
 #start building the image name

  namebuild=context #the image name starts with the context

  ###########
  #select the propic at random
  #############

  propic_path = paste0(main_path, "Propics/Round")

  propic_dirs <- dir_ls(propic_path, type = "directory")

  #We have a 95% chance of generating a white profile,
  #a 5% of generating a black one

  sel_dir <- sample(propic_dirs, 1, prob=probabilities)

  # workaround to solve the problem with accented letters and encoding.
  #Split the main path and paste it every time
  #and work only on the relevant part of the path

  split_strings <- strsplit(sel_dir, "Propics/", fixed = TRUE)[[1]]

  propics <- dir_ls(paste0(main_path, split_strings[2]), type = "file")

  #Remove the annoying google drive .ini file from the sampling
  propics= propics[grepl("desktop.ini", as.character(propics))==F]

  # Choose one file at random from these files
  propic <- sample(propics, 1)

  propic <- strsplit(propic, "Propics/", fixed = TRUE)[[1]][2]
```

```
#keep building the name including ethnicity,
#gender and age which are contained in the file name
namebuild=gsub(".png", "", paste0(namebuild, "
                                  _",
                                  gsub("/",
                                       "_",
                                       strsplit(propic,
                                                "Round/",
                                                fixed = TRUE)[[1]][2])))

#Read and scale the propic
propic <- image_read(paste0(main_path,propic)) |>
  image_scale("330")

#############
#Sample the specific characteristics
#and build the ic name
#############

temp_age=ifelse(grepl("35",namebuild)==T, "35", "70")
temp_gender=ifelse(grepl("female",namebuild)==T, "Female", "Male")
temp_ethnicity=ifelse(grepl("black",namebuild)==T, "Black", "White")
temp_job=sample(job, 1)
temp_issue=sample(issue, 1)
temp_nostalgia=sample(nostalgia, 1)
temp_valence =sample(valence, 1)
temp_food=sample(food, 1)
temp_animal=sample(animal, 1)
temp_crowd=sample(crowd, 1)

temp_name = sample(name_and_info[[context]][["Name"]][[temp_ethnicity]][[temp_gender]], 1)
temp_surname= sample(name_and_info[[context]][["Surname"]][[temp_ethnicity]][[temp_gender]], 1)

#Finally, build the name of the picture

namebuild=paste0(namebuild, "_",
                 temp_name, "_",
                 temp_surname, "_",
                 temp_job, "_",
                 temp_issue, "_",
                 temp_nostalgia, "_",
                 temp_valence, "_",
                 temp_food, "_",
                 temp_animal, "_",
                 temp_crowd, "_",
                 i,
                 # i ensures that every name is
                 # unique even if characteristics were the same
                 ".png")


#An example of a picture's name is:
# "FR_white_female_35_3_Sabrina_Bernard_Politician_rightpos_past2_corruption1_ethnicrich_dogpoor_mixe
```

```
###############################
#Now we start building all the visual blocks of the pictures
###############################

##############
############## Name and Bio
    #according to gender, ethnicity, and language
#######
##############

    ### annotatate the immage with bio, acording to gender

  #jobs, according to gender
  temp_job_context = name_and_info[[context]][["Job"]][[temp_gender]][[temp_job]]

  #the candidacy announcement, according to gender

  temp_candidacy_announcement = name_and_info[[context]][["Candidacy"]][[temp_gender]]

  #annotating bio

  temp_bio = paste0("\n",
                    temp_job_context,
                    "\n",
                    temp_candidacy_announcement
                    )


#######
#Issues
#######

issue_pic = image_read(paste0(main_path,
                              "Issue/",
                              context,
                              "/",
                              temp_issue,
                              ".png")) |>
    image_scale("480")

#####
#food
#####

food_pic = image_read(paste0(main_path,
                             "Food/",
                             temp_food,
                             ".png")) |>
    image_scale("480")

#####
#nostalgia
```

```
#####

nostalgia_pic = image_read(paste0(main_path,
                                  "Nostalgia/",
                                  context,
                                  "/",
                                  temp_nostalgia,
                                  ".png")) |>
    image_scale("480")

#####
#crowd
#####

crowd_pic = image_read(paste0(main_path,
                              "Crowd/",
                              temp_crowd,
                              ".png")) |>
    image_scale("480")

#####
#valence
#####

valence_pic = image_read(paste0(main_path,
                                "Valence/",
                                context,
                                "/",
                                temp_valence,
                                ".png")) |>
    image_scale("480")

#####
#animal
#####

animal_pic = image_read(paste0(main_path,
                               "Animal/",
                               temp_animal,
                               ".png")) |>
    image_scale("480")

#####
# Compose the finale image
#####

pic = template |>
  image_annotate(paste(temp_name,
                       temp_surname),
                 size = 60,
                 color = "Black",
                 font = "helvetica",
                 location = "+50+380"
```

```
                   ) |>
    image_annotate(temp_bio,
                   size = 50,
                   font= "helvetica",
                   color = "#392C2B",
                   location = "+50+425"
                   ) |>
    image_composite(propic,
                    offset = "+60+30") |>
    image_composite(issue_pic,
                    offset = "+0+1059") |>
    image_composite(food_pic,
                    offset = "+489+1059") |>
    image_composite(nostalgia_pic,
                    offset = "+978+1059") |>
    image_composite(crowd_pic,
                    offset = "+0+1548") |>
    image_composite(valence_pic,
                    offset = "+489+1548") |>
    image_composite(animal_pic,
                    offset = "+978+1548") |>
      image_scale("x900")

  image_write(pic, paste0(output_wd, context, "/", namebuild))

  print(paste0(i, " ",context, " ", Sys.time()))
}

}
```

Now the pictures have been generated in your desired output directory. You can randomly extract two of them to performed a forced-choice conjoint experiment in your CAWI survey. Always remember that all the information conveyed visually is contained in the name of the pictures, thus it is of the utmost importance to keep track of the names of the pictures each respondent sees in order to perform conjoint analyses.