Visual Conjoint Guide

Gaetano Scaduto

2024-07-30

# Visual Conjoint Experiments with Simulated Instagram Profiles: A Guide for Replication and explaining the code in R.

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, folders 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 (see for example [here](https://github.com/gaetanoscaduto/visual_conjoint_experiment/tree/main/example_materials/immagini/Propics/Round)).

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]](#footnote-21) To see how the directory must be organized, see an example [here](https://github.com/gaetanoscaduto/visual_conjoint_experiment/tree/main/example_materials/immagini). The content and name of the directories could be adjusted in order 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"

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](https://github.com/gaetanoscaduto/visual_conjoint_experiment/blob/main/example_materials/Names%20and%20infos.xlsx):

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  
 )  
 )  
   
}

Given the probabilities we assigned to the possibilities of seeing black vs white politicians profiles (5% vs 95%), we assign them accordingly.

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 size and proportion are based exactly on this specific template. We advise researchers who want to use this code to use the same exact picture for empty templates available [here](https://github.com/gaetanoscaduto/visual_conjoint_experiment/blob/main/example_materials/immagini/Template/template_empty1.png).

template = image\_read(paste0(main\_path,"Template/template\_empty1.png"))

for(context in contexts)  
{  
  
npics=15000 #adjust according to need  
for(i in 1:npics)  
{  
 #start building the image name  
 namebuild=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 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, "immagini/", 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, "immagini/", 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)  
   
 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")  
   
 #######  
 #Put name and bio in the pic,   
 #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()))  
}  
  
}

1. Note that if working with Google drive, you need to delete the .ini files in the “propic/round/” subfolders. [↑](#footnote-ref-21)