Attractiveness Ratings Example

Erin M. Buchanan

2023-03-28

### Vignette Setup:

knitr::opts\_chunk$set(echo = TRUE)  
  
# Set a random seed  
set.seed(5989320)  
  
# Libraries necessary for this vignette  
library(rio)  
library(flextable)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)  
library(psych)  
  
# Function for simulation  
item\_power <- function(data, # name of data frame  
 dv\_col, # name of DV column as a character  
 item\_col, # number of items column as a character  
 sample\_start = 20,   
 sample\_stop = 200,   
 sample\_increase = 5,  
 decile = .5){  
   
 DF <- cbind.data.frame(  
 "dv" = data[ , dv\_col],  
 "items" = data[ , item\_col]  
 )  
   
 # just in case  
 colnames(DF) <- c("dv", "items")  
   
 # figure out the "sufficiently narrow" ci value  
 SE <- tapply(DF$dv, DF$items, function (x) { sd(x)/sqrt(length(x)) })  
 cutoff <- quantile(SE, probs = decile)  
   
 # sequence of sample sizes to try  
 samplesize\_values <- seq(sample\_start, sample\_stop, sample\_increase)  
  
 # create a blank table for us to save the values in   
 sim\_table <- matrix(NA,   
 nrow = length(samplesize\_values),   
 ncol = length(unique(DF$items)))  
  
 # make it a data frame  
 sim\_table <- as.data.frame(sim\_table)  
  
 # add a place for sample size values   
 sim\_table$sample\_size <- NA  
  
 # loop over sample sizes  
 for (i in 1:length(samplesize\_values)){  
   
 # temp that samples and summarizes  
 temp <- DF %>%   
 group\_by(items) %>%   
 sample\_n(samplesize\_values[i], replace = T) %>%   
 summarize(se = sd(dv)/sqrt(length(dv)))  
   
 # dv on items  
 colnames(sim\_table)[1:length(unique(DF$items))] <- temp$items  
 sim\_table[i, 1:length(unique(DF$items))] <- temp$se  
 sim\_table[i, "sample\_size"] <- samplesize\_values[i]  
   
 }  
  
 # figure out cut off  
 final\_sample <- sim\_table %>%   
 pivot\_longer(cols = -c(sample\_size)) %>%   
 rename(item = name, se = value) %>%   
 group\_by(sample\_size) %>%   
 summarize(Percent\_Below = sum(se <= cutoff)/length(unique(DF$items)))   
   
 # multiply by correction   
 final\_sample$new\_sample <- round(39.369 + 0.700\*final\_sample$sample\_size + 0.003\*cutoff - 0.694\*length(unique(DF$items)))  
   
 return(list(  
 SE = SE,   
 cutoff = cutoff,   
 DF = DF,   
 sim\_table = sim\_table,   
 final\_sample = final\_sample  
 ))  
  
}

### Project/Data Title:

Attractiveness Ratings

Data provided by: Carlota Batres

### Project/Data Description:

This dataset contains 200 participants rating 20 faces on attractiveness. Ethical approval was received from the Franklin and Marshall Institutional Review Board and each participant provided informed consent. All participants were located in the United States. Participants were instructed that they would be viewing several faces which were photographed facing forward, under constant camera and lighting conditions, with neutral expressions, and closed mouths. Each participant would have to rate the attractiveness of the presented faces. More specifically, participants were asked “How attractive is this face?”, where 1 = “Not at all attractive” and 7 = “Very attractive”. Participants rated each face individually, in random order, and with no time limit. Upon completion, participants were paid for participation in the study.

### Methods Description:

The data was collected online using Amazon’s Mechanical Turk platform.

### Data Location:

Included with the vignette.

DF <- import("batres\_data.sav")  
  
str(DF)

## 'data.frame': 200 obs. of 21 variables:  
## $ Participant\_Number: num 1 2 3 4 5 6 7 8 9 10 ...  
## ..- attr(\*, "label")= chr "Unique number assigned to each participant"  
## ..- attr(\*, "format.spss")= chr "F3.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_1 : num 1 2 5 2 3 1 2 2 1 2 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #1"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_2 : num 1 6 5 2 3 1 3 2 2 2 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #2"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_3 : num 3 6 7 7 4 3 5 4 4 4 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #3"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_4 : num 3 7 5 3 4 3 3 3 4 3 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #4"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_5 : num 5 7 7 5 5 6 3 3 3 3 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #5"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_6 : num 5 5 4 5 6 5 4 4 5 3 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #6"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_7 : num 5 7 7 7 4 5 4 4 5 4 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #7"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_8 : num 4 1 5 3 4 4 4 4 2 4 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #8"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_9 : num 3 5 4 4 3 1 2 2 2 2 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #9"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_10 : num 4 4 7 2 3 3 3 3 5 4 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #10"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_11 : num 2 3 5 4 3 2 3 3 4 2 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #11"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_12 : num 4 7 5 4 4 4 3 3 6 1 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #12"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_13 : num 3 3 4 5 4 3 3 3 3 2 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #13"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_14 : num 5 7 5 5 3 5 5 5 4 2 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #14"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_15 : num 3 7 6 3 4 6 3 3 4 4 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #15"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_16 : num 4 7 5 5 5 4 4 3 5 3 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #16"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_17 : num 4 4 5 3 5 4 3 2 4 2 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #17"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_18 : num 3 5 4 6 4 5 4 5 4 2 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #18"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_19 : num 3 4 5 6 4 4 4 3 3 4 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #19"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12  
## $ Face\_20 : num 4 6 6 3 6 4 3 3 4 3 ...  
## ..- attr(\*, "label")= chr "Attractiveness rating for face #20"  
## ..- attr(\*, "format.spss")= chr "F1.0"  
## ..- attr(\*, "display\_width")= int 12

### Date Published:

No official publication date.

### Dataset Citation:

Batres, C. (2022). Attractiveness Ratings. [Data set].

### Keywords:

faces, ratings

### Use License:

Attribution-NonCommercial-ShareAlike CC BY-NC-SA

### Geographic Description - City/State/Country of Participants:

United States

### Column Metadata:

metadata <- import("batres\_metadata.xlsx")  
  
flextable(metadata) %>% autofit()

| Variable Name | Variable Description | Type (numeric, character, logical, etc.) |
| --- | --- | --- |
| Participant\_Number | Unique number assigned to each participant | Numeric |
| Face\_1 | Attractiveness rating for face #1 | Numeric |
| Face\_2 | Attractiveness rating for face #2 | Numeric |
| Face\_3 | Attractiveness rating for face #3 | Numeric |
| Face\_4 | Attractiveness rating for face #4 | Numeric |
| Face\_5 | Attractiveness rating for face #5 | Numeric |
| Face\_6 | Attractiveness rating for face #6 | Numeric |
| Face\_7 | Attractiveness rating for face #7 | Numeric |
| Face\_8 | Attractiveness rating for face #8 | Numeric |
| Face\_9 | Attractiveness rating for face #9 | Numeric |
| Face\_10 | Attractiveness rating for face #10 | Numeric |
| Face\_11 | Attractiveness rating for face #11 | Numeric |
| Face\_12 | Attractiveness rating for face #12 | Numeric |
| Face\_13 | Attractiveness rating for face #13 | Numeric |
| Face\_14 | Attractiveness rating for face #14 | Numeric |
| Face\_15 | Attractiveness rating for face #15 | Numeric |
| Face\_16 | Attractiveness rating for face #16 | Numeric |
| Face\_17 | Attractiveness rating for face #17 | Numeric |
| Face\_18 | Attractiveness rating for face #18 | Numeric |
| Face\_19 | Attractiveness rating for face #19 | Numeric |
| Face\_20 | Attractiveness rating for face #20 | Numeric |

### AIPE Analysis:

The data should be in long format with each rating on one row of data.

# Reformat the data  
DF\_long <- pivot\_longer(DF, cols = -c(Participant\_Number)) %>%   
 rename(item = name, score = value)  
  
flextable(head(DF\_long)) %>% autofit()

| Participant\_Number | item | score |
| --- | --- | --- |
| 1 | Face\_1 | 1 |
| 1 | Face\_2 | 1 |
| 1 | Face\_3 | 3 |
| 1 | Face\_4 | 3 |
| 1 | Face\_5 | 5 |
| 1 | Face\_6 | 5 |

#### Stopping Rule

# Function for simulation  
var1 <- item\_power(data = DF\_long, # name of data frame  
 dv\_col = "score", # name of DV column as a character  
 item\_col = "item", # number of items column as a character  
 sample\_start = 20,   
 sample\_stop = 300,   
 sample\_increase = 5,  
 decile = .5)

What the usual standard error for the data that could be considered for our stopping rule using the 50% decile?

# individual SEs  
var1$SE

## Face\_1 Face\_10 Face\_11 Face\_12 Face\_13 Face\_14 Face\_15   
## 0.09117808 0.09064190 0.10007472 0.09739767 0.08562437 0.08767230 0.09331351   
## Face\_16 Face\_17 Face\_18 Face\_19 Face\_2 Face\_20 Face\_3   
## 0.10262632 0.09082536 0.09530433 0.09123386 0.08818665 0.09799754 0.08644573   
## Face\_4 Face\_5 Face\_6 Face\_7 Face\_8 Face\_9   
## 0.08915009 0.09127172 0.09078109 0.09968796 0.08977638 0.09481468

var1$cutoff

## 50%   
## 0.09120597

Using our 50% decile as a guide, we find that 0.091 is our target standard error for an accurately measured item.

#### Minimum Sample Size

To estimate minimum sample size, we should figure out what number of participants it would take to achieve 80%, 85%, 90%, and 95% of the SEs for items below our critical score of 0.091?

Please note that you will always need to simulate larger than the pilot data sample size to get the starting numbers. We will correct them below. As shown in our manuscript, we need to correct for the overestimation of sample sizes based on the original pilot data size. Given that the pilot data is large: 200, this correction is especially useful. This correction is built into our function.

flextable(var1$final\_sample %>%   
 filter(Percent\_Below >= .80) %>%   
 arrange(Percent\_Below, new\_sample)) %>%   
 autofit()

| sample\_size | Percent\_Below | new\_sample |
| --- | --- | --- |
| 230 | 0.85 | 186 |
| 235 | 0.85 | 190 |
| 225 | 0.90 | 183 |
| 250 | 0.90 | 200 |
| 260 | 0.95 | 207 |
| 265 | 0.95 | 211 |
| 240 | 1.00 | 193 |
| 245 | 1.00 | 197 |
| 255 | 1.00 | 204 |
| 270 | 1.00 | 214 |
| 275 | 1.00 | 218 |
| 280 | 1.00 | 221 |
| 285 | 1.00 | 225 |
| 290 | 1.00 | 228 |
| 295 | 1.00 | 232 |
| 300 | 1.00 | 235 |

Our minimum suggested sample size does not exist at exactly 80% of the items, but instead we can use 85% (*n* = 186 as the minimum). In this example, 90% actually hits below 85% at the minimum (*n* = 183). For 95% of items hitting criterion, the minimum sample size would be approximately 207. In this case, the researcher could decide that smallest number would be their minimum sample size.

#### Maximum Sample Size

While there are many considerations for maximum sample size (time, effort, resources), the simulation suggests that 207 people would ensure nearly all items achieve cutoff criterions.

#### Final Sample Size

In any estimate for sample size, you should also consider the potential for missing data and/or unusable data due to any other exclusion criteria in your study (i.e., attention checks, speeding, getting the answer right, etc.). In this study, we likely expect all participants to see all items and therefore, we could expect to use the minimum sample size as our final sample size, the point at which all items reach our SE criterion, or the maximum sample size.