Typicality, goodness, imageability, and familiarity of stimuli across 16 categories

Necdet Gürkan & Yangyang Yu

Last Knitted: 2023-09-18

### Vignette Setup:

### Project/Data Title:

Exemplar by feature applicability matrices and other Dutch normative data for semantic concepts

Data provided by: Wolf Vanpaemel

### Project/Data Description:

This data provides extensive exemplar by feature applicability matrices covering 15 or 16 different categories (birds, fish, insects, mammals, amphibians/reptiles, clothing, kitchen utensils, musical instruments, tools, vehicles, weapons, fruit, vegetables, professions, and sports), as well as two large semantic domains (animals and artifacts). For all exemplars of the semantic categories, typicality ratings, goodness ratings, goodness rank order, generation frequency, exemplar associative strength, category associative strength, estimated age of acquisition, word frequency, familiarity ratings, imageability ratings, and pairwise similarity ratings are described as well. The structure of the dataset is not programming language friendly. Here, we only consider typicality.

### Methods Description:

The typicality data were collected as part of a larger data collection. Here we describe the typicality data collection only. The data collection took place in a large classroom where all the participants were present at the same time. The participants received a booklet with instructions on the first page, followed by four sheets with a semantic category label printed in bold on top. Each of the category labels was followed by a list of 5–33 items belonging to that category, referring to exemplars. The participants were asked to indicate, for every item in the list, how typical it was for the category printed on top of the page. They used a Likert-type rating scale, ranging from 1 for very atypical items to 20 for very typical items. If they encountered an exemplar they did not know, they were asked to circle it. Every participant completed typicality ratings for four different categories. The assignment of categories to participants was randomized. For every category, four different random permutations of the exemplars were used, and each of these permutations was distributed with an equal frequency among the participants. All the exemplars of a category were rated by 28 different participants.

### Data Location:

<https://static-content.springer.com/esm/art%3A10.3758%2FBRM.40.4.1030/MediaObjects/DeDeyne-BRM-2008b.zip> and included here.

### for typicality data -- cleaning and processing  
typicality\_fnames <- list.files(path = "data",  
 full.names = TRUE)  
  
typicality\_dfs <- lapply(typicality\_fnames, read.csv)  
  
ID <- c(1:16)  
typicality\_dfs <- mapply(cbind, typicality\_dfs, "SampleID" = ID, SIMPLIFY = F)  
  
typicality\_all\_df <- bind\_rows(typicality\_dfs)  
typicality\_all\_df\_v2 <- typicality\_all\_df %>%   
 unite("comp\_group", X:X.1, remove = TRUE) %>%   
 select(-c(30,31,32,33,34)) %>%   
 drop\_na(c(2:29)) %>%  
 filter\_all(any\_vars(!is.na(.))) %>%  
 dplyr::rename(compType = SampleID)  
# typicality\_all\_df\_v2  
typicality\_all\_df\_v3 <- typicality\_all\_df\_v2 %>%   
 select(starts\_with("X"), compType, comp\_group) %>%   
 pivot\_longer(cols = starts\_with("X"),   
 names\_to = "participant",   
 values\_to = "score")  
   
head(typicality\_all\_df\_v3)

## # A tibble: 6 × 4  
## compType comp\_group participant score  
## <int> <chr> <chr> <int>  
## 1 1 kikker\_frog X.2 18  
## 2 1 kikker\_frog X.3 20  
## 3 1 kikker\_frog X.4 19  
## 4 1 kikker\_frog X.5 12  
## 5 1 kikker\_frog X.6 20  
## 6 1 kikker\_frog X.7 15

### Date Published:

2008-11-01

### Dataset Citation:

De Deyne, S., Verheyen, S., Ameel, E. et al. Exemplar by feature applicability matrices and other Dutch normative data for semantic concepts. Behavior Research Methods 40, 1030–1048 (2008). <https://doi.org/10.3758/BRM.40.4.1030>

### Keywords:

Typicality, goodness, imageability, familiarity

### Use License:

CC-By Attribution 4.0 International

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

University of Leuven, Belgium

### Column Metadata:

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

| Variable Name | Variable Description | Type (numeric, character, logical, etc.) |
| --- | --- | --- |
| compType | Comparison type for typicality rating | Character |
| comp\_group | Individual items within compType | Character |
| participant | Participant number | Character |
| score | Typicality: how typical is the item for the category? | Numeric |

### AIPE Analysis:

#### Stopping Rule

In this example, we will pick one comparison type and use the items within that to estimate sample size. This choice is arbitrary!

# individual SEs among different comparison group  
SE <- tapply(typicality\_all\_df\_v3$score, typicality\_all\_df\_v3$compType, function (x) { sd(x)/sqrt(length(x)) })  
SE

## 1 2 3 4 5 6 7 8   
## 0.4847915 0.1868793 0.1894860 0.2326625 0.1862387 0.2310363 0.1433243 0.1751163   
## 9 10 11 12 14 16   
## 0.1888044 0.1563060 0.2512611 0.1945454 0.2042343 0.2520606

min(SE)

## [1] 0.1433243

max(SE)

## [1] 0.4847915

# comparison type 1: amphibians

typicality\_data\_gp1\_sub <- subset(typicality\_all\_df\_v3, compType == 1)  
  
# individual SEs for comparison type 1  
SE1 <- tapply(typicality\_data\_gp1\_sub$score, typicality\_data\_gp1\_sub$comp\_group, function (x) { sd(x)/sqrt(length(x)) })  
  
SE1

## kikker\_frog krokodil\_crocodile pad\_toad   
## 0.4836714 1.1085074 0.7368140   
## salamander\_salamander schildpad\_tortoise   
## 0.7531742 1.6330366

# sequence of sample sizes to try  
samplesize\_values <- seq(5, 200, 5)  
  
# create a blank table for us to save the values in   
sim\_table <- matrix(NA,   
 nrow = length(samplesize\_values),   
 ncol = length(unique(typicality\_data\_gp1\_sub$comp\_group)))  
# make it a data frame  
sim\_table <- as.data.frame(sim\_table)  
  
# add a place for sample size values   
sim\_table$sample\_size <- NA  
sim\_table$var <- "score"  
  
# loop over sample sizes for comparison type   
for (i in 1:length(samplesize\_values)){  
   
 # temp dataframe for comparison type 1 that samples and summarizes  
 temp1 <- typicality\_data\_gp1\_sub %>%   
 dplyr::group\_by(comp\_group) %>%   
 dplyr::sample\_n(samplesize\_values[i], replace = T) %>%   
 dplyr::summarize(se2 = sd(score)/sqrt(length(score)))   
   
 # add to table  
 colnames(sim\_table)[1:length(unique(typicality\_data\_gp1\_sub$comp\_group))] <- temp1$comp\_group  
 sim\_table[i, 1:length(unique(typicality\_data\_gp1\_sub$comp\_group))] <- temp1$se2  
 sim\_table[i, "sample\_size"] <- samplesize\_values[i]  
  
}

Calculate the cutoff score with information necessary for correction.

cutoff <- calculate\_cutoff(population = typicality\_data\_gp1\_sub,   
 grouping\_items = "comp\_group",  
 score = "score",   
 minimum = min(typicality\_data\_gp1\_sub$score),  
 maximum = max(typicality\_data\_gp1\_sub$score))  
  
cutoff$cutoff

## 40%   
## 0.7466301

### for response outputs   
# figure out cut off  
final\_sample <-   
 sim\_table %>%  
 pivot\_longer(cols = -c(sample\_size, var)) %>%   
 dplyr::rename(item = name, se = value) %>%   
 dplyr::group\_by(sample\_size, var) %>%   
 dplyr::summarize(percent\_below = sum(se <= cutoff$cutoff)/length(unique(typicality\_data\_gp1\_sub$comp\_group))) %>%   
 # dplyr::filter(percent\_below >= .80) %>%   
 dplyr::arrange(percent\_below) %>%   
 ungroup()

## `summarise()` has grouped output by 'sample\_size'. You can override using the  
## `.groups` argument.

flextable(final\_sample) %>% autofit()

| sample\_size | var | percent\_below |
| --- | --- | --- |
| 10 | score | 0.0 |
| 5 | score | 0.2 |
| 15 | score | 0.2 |
| 30 | score | 0.2 |
| 20 | score | 0.4 |
| 25 | score | 0.4 |
| 35 | score | 0.6 |
| 40 | score | 0.6 |
| 45 | score | 0.6 |
| 50 | score | 0.6 |
| 60 | score | 0.6 |
| 65 | score | 0.6 |
| 55 | score | 0.8 |
| 70 | score | 0.8 |
| 75 | score | 0.8 |
| 80 | score | 0.8 |
| 85 | score | 0.8 |
| 90 | score | 0.8 |
| 95 | score | 0.8 |
| 100 | score | 0.8 |
| 110 | score | 0.8 |
| 115 | score | 0.8 |
| 120 | score | 0.8 |
| 135 | score | 0.8 |
| 150 | score | 0.8 |
| 155 | score | 0.8 |
| 105 | score | 1.0 |
| 125 | score | 1.0 |
| 130 | score | 1.0 |
| 140 | score | 1.0 |
| 145 | score | 1.0 |
| 160 | score | 1.0 |
| 165 | score | 1.0 |
| 170 | score | 1.0 |
| 175 | score | 1.0 |
| 180 | score | 1.0 |
| 185 | score | 1.0 |
| 190 | score | 1.0 |
| 195 | score | 1.0 |
| 200 | score | 1.0 |

Calculate the final corrected scores:

final\_scores <- calculate\_correction(proportion\_summary = final\_sample,  
 pilot\_sample\_size = length(unique(typicality\_data\_gp1\_sub$participant)),  
 proportion\_variability = cutoff$prop\_var)  
  
flextable(final\_scores) %>% autofit()

| percent\_below | sample\_size | corrected\_sample\_size |
| --- | --- | --- |
| 80 | 55 | 53.95103 |
| 100 | 105 | 98.50778 |
| 100 | 105 | 98.50778 |
| 100 | 105 | 98.50778 |

#### Minimum Sample Size

Based on these simulations, we can decide our minimum sample size is likely close to 54.

#### Maximum Sample Size

In this example, we could set our maximum sample size for 90% power, which would equate to 99 participants.