Survival processing usefulness

Necdet Gürkan & Yangyang Yu

2023-03-28

### Vignette Setup:

### Project/Data Title:

The survival processing effect

Data provided by: Röer, Bell & Buchner (2013)

### Project/Data Description:

The data come from a conceptual replication study on the survival processing effect. The survival processing effect refers to the finding that rating words according to their relevance in a survival-related scenario leads to better retention than processing words in a number of other fictional scenarios. Participants were randomly assigned to one of the rating scenarios (survival, afterlife, moving). The to-be-rated words were presented individually in a random order on the computer screen. Each word remained on the screen for five seconds. Participants rated the words by clicking on a 5-point scale that ranged from completely useless (1) to very useful (5), which was displayed right below the word.

### Methods Description:

Participants were students at Heinrich-Heine-Universität Düsseldorf, Germany that were paid for participating or received course credit. Their ages ranged from 18 to 55 years. The words to-be-rated consisted of 30 typical members of 30 categories drawn from the updated Battig and Montague norms (Van Overschelde, Rawson, & Dunlosky, 2004).

### Data Location:

Data included within this vignette. We drop the scenario column because the standard deviation and mean of item ratings across the scenarios identical. We also add a participant column to keep this script similar to other ones.

DF <- import("roer\_data.xlsx")  
drops <- c("Scenario")  
DF <- DF[ , !(names(DF) %in% drops)]  
DF <- cbind(Participant\_Number = 1:nrow(DF) , DF)  
  
str(DF)

## 'data.frame': 218 obs. of 31 variables:  
## $ Participant\_Number: int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Item\_1 : num 1 3 3 2 3 3 3 1 4 1 ...  
## $ Item\_2 : num 2 5 4 1 4 5 4 3 5 3 ...  
## $ Item\_3 : num 5 3 1 5 2 2 5 2 3 4 ...  
## $ Item\_4 : num 1 4 2 0 3 2 2 1 4 2 ...  
## $ Item\_5 : num 3 5 1 5 1 3 5 2 0 4 ...  
## $ Item\_6 : num 1 3 2 1 4 1 5 2 2 1 ...  
## $ Item\_7 : num 4 2 4 1 4 4 4 1 5 3 ...  
## $ Item\_8 : num 3 5 5 4 4 2 5 5 2 3 ...  
## $ Item\_9 : num 1 5 4 5 4 2 4 1 3 3 ...  
## $ Item\_10 : num 0 5 5 5 0 5 5 4 0 4 ...  
## $ Item\_11 : num 3 5 4 5 3 1 5 5 2 1 ...  
## $ Item\_12 : num 3 5 5 5 5 1 3 5 1 1 ...  
## $ Item\_13 : num 1 1 3 1 2 1 1 4 1 1 ...  
## $ Item\_14 : num 1 2 1 4 1 2 2 1 2 5 ...  
## $ Item\_15 : num 1 4 2 1 5 1 3 2 2 1 ...  
## $ Item\_16 : num 4 5 4 2 3 3 4 3 3 4 ...  
## $ Item\_17 : num 3 5 0 1 3 1 4 3 3 3 ...  
## $ Item\_18 : num 2 5 4 1 5 5 5 2 5 4 ...  
## $ Item\_19 : num 2 1 1 4 1 3 4 2 4 3 ...  
## $ Item\_20 : num 5 3 4 5 2 4 5 3 3 2 ...  
## $ Item\_21 : num 3 0 4 5 3 4 1 1 4 2 ...  
## $ Item\_22 : num 3 5 1 1 2 3 5 4 3 4 ...  
## $ Item\_23 : num 1 1 2 1 1 4 3 1 5 2 ...  
## $ Item\_24 : num 1 4 4 2 3 2 4 5 2 0 ...  
## $ Item\_25 : num 2 5 5 3 5 1 5 4 3 4 ...  
## $ Item\_26 : num 1 4 3 1 3 1 2 1 2 3 ...  
## $ Item\_27 : num 1 5 1 1 5 2 4 4 3 4 ...  
## $ Item\_28 : num 1 5 1 1 4 2 3 5 2 3 ...  
## $ Item\_29 : num 1 5 2 3 2 1 4 3 1 3 ...  
## $ Item\_30 : num 3 5 4 1 1 2 5 4 4 5 ...

### Date Published:

No official publication, see citation below.

### Dataset Citation:

Röer, J. P., Bell, R., & Buchner, A. (2013). Is the survival-processing memory advantage due to richness of encoding? Journal of Experimental Psychology: Learning, Memory, and Cognition, 39, 1294-1302.

### Keywords:

usefulness; survival processing; adaptive memory; richness of encoding

### Use License:

CC BY-NC

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

Düsseldorf, Germany

### Column Metadata:

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

| Variable Name | Variable Description | Type (numeric, character, logical, etc.) |
| --- | --- | --- |
| Items | Item ratings for item\_1 to item\_30 | Numeric |
| Scenario | Categorical scenarios- 1,2,3 | Numeric |

### AIPE Analysis:

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

| Participant\_Number | item | score |
| --- | --- | --- |
| 1 | Item\_1 | 1 |
| 1 | Item\_2 | 2 |
| 1 | Item\_3 | 5 |
| 1 | Item\_4 | 1 |
| 1 | Item\_5 | 3 |
| 1 | Item\_6 | 1 |

#### Stopping Rule

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

# individual SEs  
SE <- tapply(DF\_long$score, DF\_long$item, function (x) { sd(x)/sqrt(length(x)) })  
  
SE

## Item\_1 Item\_10 Item\_11 Item\_12 Item\_13 Item\_14 Item\_15   
## 0.08860091 0.09743922 0.10419876 0.10725788 0.07167329 0.11502337 0.09746261   
## Item\_16 Item\_17 Item\_18 Item\_19 Item\_2 Item\_20 Item\_21   
## 0.09425595 0.08981453 0.08968434 0.09817041 0.10421504 0.10768420 0.10195369   
## Item\_22 Item\_23 Item\_24 Item\_25 Item\_26 Item\_27 Item\_28   
## 0.09891678 0.11403914 0.08751293 0.10279479 0.08015419 0.09612067 0.09697587   
## Item\_29 Item\_3 Item\_30 Item\_4 Item\_5 Item\_6 Item\_7   
## 0.08758437 0.10963101 0.11319421 0.07944272 0.11514133 0.08852481 0.10163361   
## Item\_8 Item\_9   
## 0.09615698 0.09118122

cutoff <- quantile(SE, probs = .50)  
cutoff

## 50%   
## 0.09745092

Using our 50% decile as a guide, we find that 0.097 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 0.097?

# sequence of sample sizes to try  
samplesize\_values <- seq(20, 300, 5)  
  
# create a blank table for us to save the values in   
sim\_table <- matrix(NA,   
 nrow = length(samplesize\_values),   
 ncol = length(unique(DF\_long$item)))  
  
# 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 dataframe that samples and summarizes  
 temp <- DF\_long %>%   
 group\_by(item) %>%   
 sample\_n(samplesize\_values[i], replace = T) %>%   
 summarize(se = sd(score)/sqrt(length(score)))   
   
 colnames(sim\_table)[1:length(unique(DF\_long$item))] <- temp$item  
 sim\_table[i, 1:length(unique(DF\_long$item))] <- temp$se  
 sim\_table[i, "sample\_size"] <- samplesize\_values[i]  
 }  
  
final\_sample <-   
 sim\_table %>%   
 pivot\_longer(cols = -c(sample\_size)) %>%   
 dplyr::rename(item = name, se = value) %>%   
 group\_by(sample\_size) %>%   
 summarize(Percent\_Below = sum(se <= cutoff)/length(unique(DF\_long$item))) %>%   
 filter(Percent\_Below >= .80) %>%   
 mutate(new\_sample = round(39.369 + 0.700\*sample\_size + 0.003\*cutoff - 0.694\*length(unique(DF\_long$item))))  
  
flextable(final\_sample) %>% autofit()

| sample\_size | Percent\_Below | new\_sample |
| --- | --- | --- |
| 265 | 0.8000000 | 204 |
| 270 | 0.8333333 | 208 |
| 275 | 0.8333333 | 211 |
| 280 | 0.8666667 | 215 |
| 285 | 0.8333333 | 218 |
| 290 | 0.9333333 | 222 |
| 295 | 0.8666667 | 225 |
| 300 | 0.9000000 | 229 |

Based on these simulations, we can decide our minimum sample size is likely close to 204.

#### Maximum Sample Size

In this example, we could set our maximum sample size for 90% power, which would equate to 222 participants.

#### Final Sample Size

We should consider any data loss or other issues related to survival processing when thinking about the final sample size requirements.