Reaction Time Example (raw RT)

Christopher L. Aberson

2023-03-28

### Vignette Setup:

### Project/Data Title:

Speeded word fragment completion. Classification of Dutch words as either actual words or nonwords

Data provided by: Tom Heyman

### Project/Data Description:

The data come from a study reported in Heyman, De Deyne, Hutchison, & Storms (2015, Behavior Research Methods; henceforth HDHS). More specifically, the study involved a continuous lexical decision task intended to measure (item-level) semantic priming effects (i.e., Experiment 3 of HDHS). It is similar to the SPAML set-up, but with fewer items and participants. The study had several goals, but principally we wanted to examine how a different/new paradigm called the speeded word fragment completion task would compare against a more common, well-established paradigm like lexical decision in terms of semantic priming (i.e., magnitude of the effect, reliability of item-level priming, cross-task correlation of item-level priming effects,…). Experiment 3 only involved a continuous lexical decision task, so the datafile contains no data from the speeded word fragment completion task (I can share those as well, if useful).

### Methods Description:

Participants were 40 students of the University of Leuven, Belgium (10 men, 30 women, mean age 20 years). A total of 576 pairs were used in a continuous lexical decision task (so participants don’t perceive them as pairs): 144 word–word pairs, 144 word–pseudoword pairs, 144 pseudoword–word pairs, and 144 pseudoword–pseudoword pairs. Of the 144 word-word pairs 72 were fillers, and 72 were critical pairs, half of which were related, the other half unrelated (this was counterbalanced across participants). The dataset only contains data for the critical pairs. Participants were informed that they would see a letter string on each trial and that they had to indicate whether the letter string formed an existing Dutch word or not by pressing the arrow keys. Half of the participants had to press the left arrow for word and the right arrow for nonword, and vice versa for the other half.

### Data Location:

<https://osf.io/frxpd/> [also contains R code with some AIPE implementation]

HDHS<- read.csv("HDHSAIPE.txt", sep="")  
str(HDHS)

## 'data.frame': 2880 obs. of 8 variables:  
## $ RT : num 0.52 0.453 0.467 0.534 0.573 ...  
## $ zRT : num -0.303 -0.492 -0.453 -0.265 -0.153 ...  
## $ Pp : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Type : chr "R" "R" "R" "R" ...  
## $ Prime : chr "hengst" "matrak" "eland" "erwt" ...  
## $ Target : chr "veulen" "wapen" "gewei" "wortel" ...  
## $ accTarget: int 1 1 1 1 1 1 1 1 1 0 ...  
## $ accPrime : int 1 1 1 1 0 1 1 1 1 0 ...

### Date Published:

2022-02-04

### Dataset Citation:

Heyman, T. (2022, February 4). Dataset AIPE. Retrieved from osf.io/frxpd [based on Heyman, T., De Deyne, S., Hutchison, K. A., & Storms, G. (2015). Using the speeded word fragment completion task to examine semantic priming. Behavior Research Methods, 47(2), 580-606.]

### Keywords:

Semantic priming; continuous lexical decision task

### Use License:

CC-By Attribution 4.0 International

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

Belgium

### Column Metadata:

metadata <- import("HDHSMeta.txt")  
  
flextable(metadata) %>% autofit()

| Variable Name | Variable Description | Type (numeric, character, logical, etc.) |
| --- | --- | --- |
| RT | Response time to the target in seconds | Numeric |
| zRT | Z-transformed target response times per participant | Numeric |
| Pp | Participant identifier (1 to 40) | Integer |
| Type | Whether target was preceded by a related prime (<93>R<94>) or an unrelated prime (<93>U<94>) | Character |
| Prime | Prime stimulus (in Dutch) | Character |
| Target | Target stimulus (in Dutch) | Character |
| accTarget | Whether response to target was correct (1) or not (0) | Integer |
| accPrime | Whether response to the preceding prime was correct (1) or not (0) | Integer |

### AIPE Analysis:

HDHScorrect <- HDHS[HDHS$accTarget==1,]   
summary\_stats <- HDHScorrect %>% #data frame  
 select(RT, Target) %>% #pick the columns  
 group\_by(Target) %>% #put together the stimuli  
 summarize(SES = sd(RT)/sqrt(length(RT)), samplesize = length(RT)) #create SE and the sample size for below  
##give descriptives of the SEs  
describe(summary\_stats$SES)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 72 0.05 0.05 0.03 0.04 0.02 0.02 0.31 0.29 3.63 13.19 0.01

##figure out the original sample sizes (not really necessary as all Primes were seen by 40 participants)  
original\_SS <- HDHS %>% #data frame  
 count(Target) #count up the sample size  
##add the original sample size to the data frame  
summary\_stats <- merge(summary\_stats, original\_SS, by = "Target")  
##original sample size average  
describe(summary\_stats$n)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 72 40 0 40 40 0 40 40 0 NaN NaN 0

##reduced sample size  
describe(summary\_stats$samplesize)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 72 38.12 3.09 39 38.83 1.48 22 40 18 -3.29 12.08 0.36

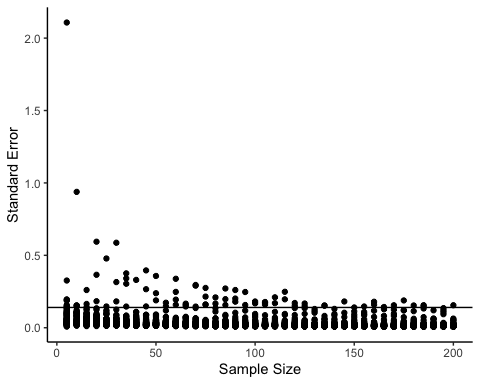
##percent retained  
describe(summary\_stats$samplesize/summary\_stats$n)

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 72 0.95 0.08 0.98 0.97 0.04 0.55 1 0.45 -3.29 12.08 0.01

##average SE for words with at least n = 30  
summary\_stats %>% #data frame  
 filter(samplesize >=30) %>% #filter out lower sample sizes  
 summarize(avgSES = mean(SES)) #create the mean

## avgSES  
## 1 0.04766927

##pick all words with sample sizes above 30  
targets <- summary\_stats %>% #data frame  
 filter(samplesize >=30) %>% #filter out sample sizes  
 pull(Target) #return a vector  
targets <- as.character(targets)  
  
##this section creates a sequence of sample sizes to estimate at  
#5, 10, 15, etc.  
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(targets))  
#create column names based on the current targets  
colnames(sim\_table) <- targets  
#make it a data frame  
sim\_table <- as.data.frame(sim\_table)  
#add those sample size values  
sim\_table$sample\_size <- samplesize\_values  
##loop over all the target words randomly selected  
for (i in 1:length(targets)){  
 ##loop over sample sizes  
 for (q in 1:length(samplesize\_values)){  
 ##temporarily save a data frame of Zscores  
 temp <- HDHScorrect %>% #data frame  
 filter(Target == targets[i]) %>% #pick rows that are the current target word  
 sample\_n(samplesize\_values[q], replace = T) %>% #select sample size number of rows  
 pull(RT)  
 #put that in the table  
 #find the sample size row and column we are working with  
 #calculate SE sd/sqrt(n)  
 sim\_table[sim\_table$sample\_size == samplesize\_values[q], targets[i]] <- sd(temp)/sqrt(length(temp))  
 }  
}  
  
##melt down the data into long format for ggplot2  
sim\_table\_long <- melt(sim\_table,  
 id = "sample\_size")  
##create a graph of the sample size by SE value  
ggplot(sim\_table\_long, aes(sample\_size, value)) +  
 theme\_classic() +  
 xlab("Sample Size") +  
 ylab("Standard Error") +  
 geom\_point() +  
 geom\_hline(yintercept = .14) #mark here .14 occurs



flextable(head(HDHScorrect)) %>% autofit()

| RT | zRT | Pp | Type | Prime | Target | accTarget | accPrime |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0.5202093 | -0.3026836 | 1 | R | hengst | veulen | 1 | 1 |
| 0.4532606 | -0.4918172 | 1 | R | matrak | wapen | 1 | 1 |
| 0.4670391 | -0.4528923 | 1 | R | eland | gewei | 1 | 1 |
| 0.5335296 | -0.2650529 | 1 | R | erwt | wortel | 1 | 1 |
| 0.5732744 | -0.1527717 | 1 | R | ijzel | glad | 1 | 0 |
| 0.3870141 | -0.6789673 | 1 | R | sauna | warm | 1 | 1 |

#### Stopping Rule

What the usual standard error for the data that could be considered for our stopping rule?

SE <- tapply(HDHScorrect$RT, HDHScorrect$Prime, function (x) { sd(x)/sqrt(length(x)) })  
min(SE)

## [1] 0.01510663

max(SE)

## [1] 0.3721617

cutoff <- quantile(SE, probs = .5)

The items have a range of 0.0151066 to 0.3721617. We could use the 50% decile SE = 0.0322288 as our critical value for our stopping rule, as suggested by the manuscript analysis. We could decide to be more conservative if the pilot data was smaller by using the third quartile or the 90th percentile SE. (Note: you could use smaller values of SE as well, but in general, it may pay off to be more conservative on the sample size estimates). We could also have a set SE to a specific target if we do not believe we have representative pilot data in this example. You should also consider the scale when estimating these values (i.e., 1-7 scales will have smaller estimates than 1-100 scales).

#### Minimum Sample Size

To estimate minimum sample size, we should figure out what number of participants it would take to achieve 80% of the SEs for items below our critical score of 0.0322288?

# sequence of sample sizes to try  
samplesize\_values <- seq(20, 200, 5)  
  
# create a blank table for us to save the values in   
  
sim\_table <- matrix(NA,   
 nrow = length(samplesize\_values),   
 ncol = length(unique(HDHS$Prime)))  
  
# 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 <- HDHScorrect %>%   
 group\_by(Prime) %>%   
 sample\_n(samplesize\_values[i], replace = T) %>%   
 summarize(se = sd(RT)/sqrt(length(RT)))   
   
 colnames(sim\_table)[1:length(unique(HDHScorrect$Prime))] <- temp$Prime  
 sim\_table[i, 1:length(unique(HDHScorrect$Prime))] <- temp$se  
 sim\_table[i, "sample\_size"] <- samplesize\_values[i]  
 }  
  
final\_sample <-   
 sim\_table %>%   
 pivot\_longer(cols = -c(sample\_size)) %>%   
# rename(item = variable, se = value) %>%   
 group\_by(sample\_size) %>%   
 summarize(Percent\_Below = sum(value <= cutoff)/length(unique(HDHScorrect$Prime))) %>%   
 filter(Percent\_Below >= .80)  
  
 # multiply by correction   
 final\_sample$new\_sample <- round(39.369 + 0.700\*final\_sample$sample\_size + 0.003\*cutoff - 0.694\*length(unique(HDHScorrect$Prime)))  
  
flextable(final\_sample) %>% autofit()

| sample\_size | Percent\_Below | new\_sample |
| --- | --- | --- |
| 95 | 0.8472222 | 56 |
| 105 | 0.8194444 | 63 |
| 110 | 0.8055556 | 66 |
| 115 | 0.8472222 | 70 |
| 120 | 0.8194444 | 73 |
| 125 | 0.8750000 | 77 |
| 130 | 0.8750000 | 80 |
| 135 | 0.8750000 | 84 |
| 140 | 0.8472222 | 87 |
| 145 | 0.8750000 | 91 |
| 150 | 0.8888889 | 94 |
| 155 | 0.8750000 | 98 |
| 160 | 0.8750000 | 101 |
| 165 | 0.8888889 | 105 |
| 170 | 0.8888889 | 108 |
| 175 | 0.9027778 | 112 |
| 180 | 0.8750000 | 115 |
| 185 | 0.9027778 | 119 |
| 190 | 0.8888889 | 122 |
| 195 | 0.8888889 | 126 |
| 200 | 0.8888889 | 129 |

#### Expected Data Loss

The original data found that 4.6875 percent of the data were unusable.

#### Minimum Sample Size

Based on these simulations, we can decide our minimum sample size is likely close to 59 including information about data loss. This value seems low, so we could use a higher criterion at 90% *n* = 112.

#### Maximum Sample Size

In this example, we could set our maximum sample size for 95% power, which would equate to . participants for 94% power.

#### 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. Note that maximum sample sizes can also be defined by time, money, or other means.