PACT Experiment 1A

Shawn Cummings and Rachel Theodore

/today

## Data import and counter-balance check

First, we import all spreadsheets for all conditions intended to be run. We add the condition variable to each, for identifiability once they are joined.

For each spreadsheet, we additionally check against our counterbalancing variables for that experiment and condition.

d.001 <- read.csv("../Spreadsheets/PACT-1a-001.csv", na.strings=c("", " ", "NA")) %>%  
 mutate(condition = "1a-001")  
unique(d.001$dose) #check that Dose, Test, Bias.Joanne, and Button1 are all correct

## [1] "1x"

#d.002 <- read.csv(")  
  
  
# conditions <- c("001", "002", "003", "004", "005", "006", "007", "008")  
# dose <- c("1x")  
# test.Type <- c("Blocked")  
# test.Button1 <- c("# ASI", "# ASI", "# ASI", "# ASI", "# ASHI", "# ASHI", "# ASHI", "# ASHI")  
# bias.Joanne <- c("SH", "SH", "SS", "SS", "SH", "SH", "SS", "SS")  
# first.Test.Block <- c("Peter", "Joanne", "Peter", "Joanne", "Peter", "Joanne", "Peter", "Joanne")  
#   
# d.counterbalances <- data.frame(conditions, dose, test.Type, test.Button1, bias.Joanne, first.Test.Block)  
  
  
  
  
#d <- rbind(d.001, d.002, etc.)  
d <- d.001  
  
  
exposure.Trials <- d %>%  
 filter(phase == "Exposure") %>%  
 nrow()  
  
test.Trials <- d %>%  
 filter(phase == "Test") %>% #works differently for blocked versus mixed test  
 nrow()  
  
exposure.Trials

## [1] 80

## Additional columns of interest

Using a parse of audio file used, we generate a number of columns informative towards condition. These *should* be redundant with hard-coded columns from the spreadsheet, but are included as a check against anything being awry.

d %<>%  
 separate(  
 audio, c("audio.Construction", "audio.WordType", "audio.number", "audio.word"), sep = "\_", remove=FALSE) %>%  
 mutate(  
 bias =  
 case\_when(  
 fricative == "SS" & type == "Clear" ~ "SH",  
 fricative == "SH" & type == "Ambiguous" ~ "SH",  
 fricative == "SS" & type == "Ambiguous" ~ "SS",  
 fricative == "SH" & type == "Clear" ~ "SS",  
 T ~ NA\_character\_),  
 type.Coded =  
 case\_when(  
 substr(audio.WordType, 1, 1) == "X" ~ "Ambiguous",  
 substr(audio.WordType, 1, 1) == "s" ~ "Clear",  
 audio.WordType == "Test" ~ "Test",  
 audio.WordType == "FW" ~ "Filler",  
 T ~ NA\_character\_),  
 fricative.Coded =   
 case\_when(  
 audio.WordType %in% c("Xsh", "sh") ~ "SH",  
 audio.WordType %in% c("Xs", "s") ~ "SS",  
 T ~ NA\_character\_),  
 talker.Coded =   
 case\_when(  
 substr(audio.Construction, 1, 2) == "f1" ~ "Joanne",  
 substr(audio.Construction, 1, 2) == "pe" ~ "Peter",  
 T ~ NA\_character\_))

### Check for mismatches

Return any rows where the autocoded versus manually entered values for type, talker, or fricative don’t match. This *should* return 0 rows.

d %>%  
 mutate(coding.Mismatch =  
 case\_when(  
 is.na(audio) ~ F,  
 type == type.Coded &  
 talker == talker.Coded &  
 (fricative == fricative.Coded |  
 (is.na(fricative) &  
 is.na(fricative.Coded))) ~ F,  
 T ~ T)) %>%  
 filter(coding.Mismatch == T) %>%  
 select(condition, audio,  
 type, type.Coded,  
 talker, talker.Coded,  
 fricative,fricative.Coded,  
 coding.Mismatch)

## [1] condition audio type type.Coded   
## [5] talker talker.Coded fricative fricative.Coded  
## [9] coding.Mismatch  
## <0 rows> (or 0-length row.names)