DataPrepStressDeafness

2025-05-30

The following file includes:

- data cleaning
- initial analyses before exclusion of participants
- initial analyses of all types of contrasts (voicing, vowel, stress)
- a model on X=A and X=B ('correct response'), as this was calculated on all types of trials (not just stress contrast)
- preparation of data when analysis must be done on a subset (e.g. only stress items)

Data Cleaning

Steps to data cleaning

- download all complete files and combine using ED
- take out unnecessary columns manually in Excel

read in simplified results file

```
table = read.delim ("data/ABXOrigName.txt", stringsAsFactors=TRUE)
number.of.participants = 15 #without any exclusions, anyone who has completed
options (width = 200)
head(table,10)
```

```
##    subject Version listStatus LexTale testABX.index testABX.Item testABX.Correct testABX.Xtype testAl
## 1   929263       1                 NA            NA           NA
## 2   929263       1                 NA            NA           NA
## 3   929263       1                 NA            NA           NA
## 4   929263       1                 NA            NA           NA
## 5   929263       1                 NA            NA           NA
## 6   929263       1                 NA            NA           NA
## 7   929263       1   PREQUEST      NA            NA           NA
## 8   929263       1   PREQUEST      NA            NA           NA
## 9   929263       1                 NA            NA           NA
## 10  929263       1   PREQUEST      NA            NA           NA
```

```
tail(table,10)
```

```
##      subject Version listStatus LexTale testABX.index testABX.Item testABX.Correct testABX.Xtype tes
## 2274  123572       1    ABXTEST   76.25            43           43               A    irrelevant
## 2275  123572       1    ABXTEST   76.25            44           21               A     ambiguous
## 2276  123572       1    ABXTEST   76.25            45           16               A     ambiguous
## 2277  123572       1    ABXTEST   76.25            46           31               B    irrelevant
## 2278  123572       1    ABXTEST   76.25            47           19               B     ambiguous
## 2279  123572       1    ABXTEST   76.25            48           11               B       extreme
## 2280  123572       1                 NA            NA           NA
## 2281  123572       1  POSTQUEST      NA            NA           NA
## 2282  123572       1  POSTQUEST      NA            NA           NA
## 2283  123572       1  POSTQUEST      NA            NA           NA
```

use coding to take out some unnecessary rows

```
#for one criterion, use following code:
#table = subset(table, listStatus != 'PREQUEST')

table<-subset(table, listStatus != 'PREQUEST' & listStatus != 'LEXTALE' & listStatus != 'TEST1' & listS

# list of criteria | 'LEXTALE' | 'TEST1' | 'SORTING1' | 'SORTING2A' | 'SORTING2B' | 'POSTQUEST'
head(table,10)
```

```
##     subject Version listStatus LexTale testABX.index testABX.Item testABX.Correct testABX.Xtype testA
## 111  929263       1    ABXTEST   71.25             1           44               A    irrelevant
## 112  929263       1    ABXTEST   71.25             2            5               A       extreme
## 113  929263       1    ABXTEST   71.25             3           48               A    irrelevant
## 114  929263       1    ABXTEST   71.25             4            2               B       extreme
## 115  929263       1    ABXTEST   71.25             5           46               A    irrelevant
## 116  929263       1    ABXTEST   71.25             6           47               B    irrelevant
## 117  929263       1    ABXTEST   71.25             7           32               A    irrelevant
## 118  929263       1    ABXTEST   71.25             8           27               A    irrelevant
## 119  929263       1    ABXTEST   71.25             9           39               A    irrelevant
## 120  929263       1    ABXTEST   71.25            10           30               A    irrelevant
```

```
tail(table,10)
```

```
##      subject Version listStatus LexTale testABX.index testABX.Item testABX.Correct testABX.Xtype test
## 2270  123572       1    ABXTEST   76.25            39            4               A       extreme
## 2271  123572       1    ABXTEST   76.25            40           48               A    irrelevant
## 2272  123572       1    ABXTEST   76.25            41           18               B     ambiguous
## 2273  123572       1    ABXTEST   76.25            42           23               B     ambiguous
## 2274  123572       1    ABXTEST   76.25            43           43               A    irrelevant
## 2275  123572       1    ABXTEST   76.25            44           21               A     ambiguous
## 2276  123572       1    ABXTEST   76.25            45           16               A     ambiguous
## 2277  123572       1    ABXTEST   76.25            46           31               B    irrelevant
## 2278  123572       1    ABXTEST   76.25            47           19               B     ambiguous
## 2279  123572       1    ABXTEST   76.25            48           11               B       extreme
```

read out for further manual cleaning

```
write.table(table, file="data/SDCleaned.txt", quote = FALSE, row.names = FALSE, sep = "\t")
```

- remove lisStatus column using Excel
- rename columns as needed (See Appendix A of paper for how columns are renamed)
- save as SDCleanedRenamed.txt note: added one participant who completed after first cleaning steps

read in simplified results file

```
table = read.delim ("data/SDCleanedRenamed.txt", stringsAsFactors=TRUE)
head(table,10)
```

```
##    participant training LexTale trialIndex item correct      Xtype contrast pair response accuracy
## 1       929263        1   71.25          1   44       A irrelevant  voicing    8        B        0
## 2       929263        1   71.25          2    5       A    extreme   stress    2        A        1
## 3       929263        1   71.25          3   48       A irrelevant  voicing    9        B        0
## 4       929263        1   71.25          4    2       B    extreme   stress    1        B        1
## 5       929263        1   71.25          5   46       A irrelevant  voicing    8        A        1
## 6       929263        1   71.25          6   47       B irrelevant  voicing    9        A        0
## 7       929263        1   71.25          7   32       A irrelevant    vowel    6        A        1
## 8       929263        1   71.25          8   27       A irrelevant    vowel    5        B        0
```

```
## 9       929263        1   71.25          9   39       A irrelevant  voicing    9        B        0
## 10      929263        1   71.25         10   30       A irrelevant    vowel    6        A        1
```

```
tail(table,10)
```

```
##     participant training LexTale trialIndex item correct      Xtype contrast pair response accuracy
## 701      162417        1   76.25         39   21       A  ambiguous   stress    3        A        1
## 702      162417        1   76.25         40   17       A  ambiguous   stress    2        A        1
## 703      162417        1   76.25         41   33       A irrelevant  voicing    7        B        0
## 704      162417        1   76.25         42   24       A  ambiguous   stress    3        B        0
## 705      162417        1   76.25         43   42       B irrelevant  voicing    7        A        0
## 706      162417        1   76.25         44   22       B  ambiguous   stress    3        B        1
## 707      162417        1   76.25         45   20       A  ambiguous   stress    2        B        0
## 708      162417        1   76.25         46   29       A irrelevant    vowel    5        B        0
## 709      162417        1   76.25         47   47       B irrelevant  voicing    9       -1       -1
## 710      162417        1   76.25         48    8       A    extreme   stress    2        A        1
```

Columns have now been renamed as the relevant predictors/outcomes

renaming and reorganizing for clarity

add letters to columns so they are not treated as integers

```
table$participant <- sub("^", "p", table$participant)
table$item <- sub("^", "i", table$item)
table$participant <- as.factor(table$participant)
table$trialIndex <- as.factor(table$trialIndex)
table$item <- as.factor(table$item)
```

replace the training and pairs with sensical, non-numerical labels

```
table$training <- sub("1", "stress", table$training)
table$training <- sub("2", "nonStress", table$training)
table$training <- as.factor(table$training)
table$pair <- sub("1", "refund", table$pair)
table$pair <- sub("2", "extract", table$pair)
table$pair <- sub("3", "conduct", table$pair)
table$pair <- sub("4", "list", table$pair)
table$pair <- sub("5", "living", table$pair)
table$pair <- sub("6", "litter", table$pair)
table$pair <- sub("7", "device", table$pair)
table$pair <- sub("8", "faces", table$pair)
table$pair <- sub("9", "dose", table$pair)
table$pair <- as.factor(table$pair)
head(table,10)
```

```
##    participant training LexTale trialIndex item correct      Xtype contrast    pair response accuracy
## 1      p929263   stress   71.25          1  i44       A irrelevant  voicing   faces        B        0
## 2      p929263   stress   71.25          2   i5       A    extreme   stress extract        A        1
## 3      p929263   stress   71.25          3  i48       A irrelevant  voicing    dose        B        0
## 4      p929263   stress   71.25          4   i2       B    extreme   stress  refund        B        1
## 5      p929263   stress   71.25          5  i46       A irrelevant  voicing   faces        A        1
## 6      p929263   stress   71.25          6  i47       B irrelevant  voicing    dose        A        0
## 7      p929263   stress   71.25          7  i32       A irrelevant    vowel  litter        A        1
## 8      p929263   stress   71.25          8  i27       A irrelevant    vowel  living        B        0
## 9      p929263   stress   71.25          9  i39       A irrelevant  voicing    dose        B        0
## 10     p929263   stress   71.25         10  i30       A irrelevant    vowel  litter        A        1
```

```
tail(table,10)
```

```
##     participant training LexTale trialIndex item correct      Xtype contrast    pair response accura
## 701     p162417   stress   76.25         39  i21       A  ambiguous   stress conduct        A
## 702     p162417   stress   76.25         40  i17       A  ambiguous   stress extract        A
## 703     p162417   stress   76.25         41  i33       A irrelevant  voicing  device        B
## 704     p162417   stress   76.25         42  i24       A  ambiguous   stress conduct        B
## 705     p162417   stress   76.25         43  i42       B irrelevant  voicing  device        A
## 706     p162417   stress   76.25         44  i22       B  ambiguous   stress conduct        B
## 707     p162417   stress   76.25         45  i20       A  ambiguous   stress extract        B
## 708     p162417   stress   76.25         46  i29       A irrelevant    vowel  living        B
## 709     p162417   stress   76.25         47  i47       B irrelevant  voicing    dose       -1
## 710     p162417   stress   76.25         48   i8       A    extreme   stress extract        A
```

```
levels(table$participant)
```

```
##  [1] "p123572" "p146214" "p162417" "p188768" "p34258"  "p463319" "p466179" "p491113" "p603448" "p731
```

Basic analysis for descriptive statistics

remove rows in which participant did not give answer (timed out)

```
table = subset(table, accuracy != -1)
#table
head(table,10)
```

```
##    participant training LexTale trialIndex item correct      Xtype contrast    pair response accuracy
## 1      p929263   stress   71.25          1  i44       A irrelevant  voicing   faces        B        0
## 2      p929263   stress   71.25          2   i5       A    extreme   stress extract        A        1
## 3      p929263   stress   71.25          3  i48       A irrelevant  voicing    dose        B        0
## 4      p929263   stress   71.25          4   i2       B    extreme   stress  refund        B        1
## 5      p929263   stress   71.25          5  i46       A irrelevant  voicing   faces        A        1
## 6      p929263   stress   71.25          6  i47       B irrelevant  voicing    dose        A        0
## 7      p929263   stress   71.25          7  i32       A irrelevant    vowel  litter        A        1
## 8      p929263   stress   71.25          8  i27       A irrelevant    vowel  living        B        0
## 9      p929263   stress   71.25          9  i39       A irrelevant  voicing    dose        B        0
## 10     p929263   stress   71.25         10  i30       A irrelevant    vowel  litter        A        1
```

```
tail(table,10)
```

```
##     participant training LexTale trialIndex item correct      Xtype contrast    pair response accura
## 700     p162417   stress   76.25         38  i26       B irrelevant    vowel    list        B
## 701     p162417   stress   76.25         39  i21       A  ambiguous   stress conduct        A
## 702     p162417   stress   76.25         40  i17       A  ambiguous   stress extract        A
## 703     p162417   stress   76.25         41  i33       A irrelevant  voicing  device        B
## 704     p162417   stress   76.25         42  i24       A  ambiguous   stress conduct        B
## 705     p162417   stress   76.25         43  i42       B irrelevant  voicing  device        A
## 706     p162417   stress   76.25         44  i22       B  ambiguous   stress conduct        B
## 707     p162417   stress   76.25         45  i20       A  ambiguous   stress extract        B
## 708     p162417   stress   76.25         46  i29       A irrelevant    vowel  living        B
## 710     p162417   stress   76.25         48   i8       A    extreme   stress extract        A
```

Do a quick analysis by item to verify none is an extreme outlier

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------------------------------------------
```

```
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ---------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
itemAcc = table %>% group_by(item) %>% summarise(acc = mean(accuracy), sd = sd(accuracy))
itemAcc
```

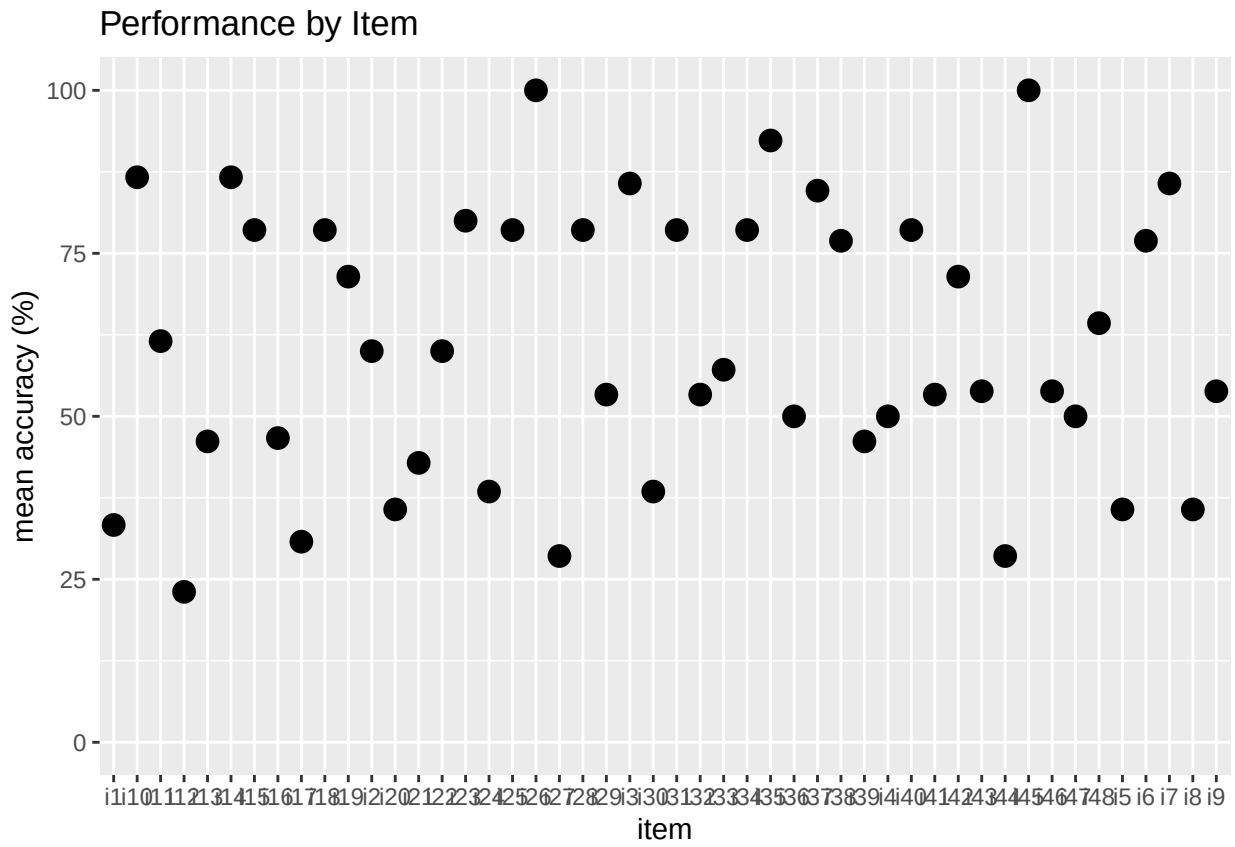
```
## # A tibble: 48 x 3
##    item    acc    sd
##    <fct> <dbl> <dbl>
##  1 i1    0.333 0.488
##  2 i10   0.867 0.352
##  3 i11   0.615 0.506
##  4 i12   0.231 0.439
##  5 i13   0.462 0.519
##  6 i14   0.867 0.352
##  7 i15   0.786 0.426
##  8 i16   0.467 0.516
##  9 i17   0.308 0.480
## 10 i18   0.786 0.426
## # i 38 more rows
```

```
library(ggplot2)
p <- ggplot(data = itemAcc, aes(x= item, y = acc*100))+

    ## defining plot type and y-axis variable

    geom_dotplot(binaxis= "y", stackdir = "center", stackgroups = TRUE) +

  ylim(0,100) +

  labs(y= "mean accuracy (%)", title = "Performance by Item")
```

```
## `geom_dotplot()` called with `stackgroups = TRUE` and `method = "dotdensity"`.", i = "Do you want `bi
```

```
p
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with `binwidth`.
```



calculate how often each participant responded A or B

```
keytable = table %>% group_by(participant) %>% count(response == "A") #getting basic data from the main
keytable = keytable %>% rename_at ('response == "A"', ~'Apressed') %>% spread(Apressed, n)
keytable = keytable %>% rename_at ('FALSE', ~'B')
keytable = keytable %>% rename_at ('TRUE', ~'A')
keytable = keytable %>% mutate ( BfreqPerc = (B/(A+B))*100) # add column for the percent of B answers g
keytable
```

```
## # A tibble: 15 x 4
## # Groups:   participant [15]
##    participant     B     A BfreqPerc
##    <fct>       <int> <int>     <dbl>
##  1 p123572        22    20      52.4
##  2 p146214        24    24      50
##  3 p162417        28    19      59.6
##  4 p188768        25    12      67.6
##  5 p34258         31    16      66.0
##  6 p463319        37     5      88.1
##  7 p466179        34    13      72.3
##  8 p491113        26    21      55.3
##  9 p603448        41     7      85.4
## 10 p731581        31    17      64.6
## 11 p874028        33    15      68.8
## 12 p929263        26    21      55.3
## 13 p960446        41     3      93.2
## 14 p965925        36    12      75
```

```
## 15 p995032        14    14      50
```

The above reflects that p463319, p603448, and p960446 almost always answered B

show the means for each contrast by participant (useful for verifying exclusion by performance)

```
strM = table %>% filter(contrast == "stress") %>% group_by(participant) %>% summarise(acc = mean(accura
voiM = table %>% filter(contrast == "voicing") %>% group_by(participant) %>% summarise(acc = mean(accur
vowM = table %>% filter(contrast == "vowel") %>% group_by(participant) %>% summarise(acc = mean(accurac
meansCbyPar = data.frame(rbind(strM, voiM, vowM))
meansCbyPar$contrast = rep(c("stress","voicing", "vowel"), each = number.of.participants) #adding a lab
meansCbyPar %>% arrange(participant)
```

```
##    participant       acc        sd contrast
## 1       p123572 0.2727273 0.4558423   stress
## 2       p123572 0.5000000 0.5188745  voicing
## 3       p123572 0.3333333 0.5163978    vowel
## 4       p146214 0.6250000 0.4945354   stress
## 5       p146214 0.7500000 0.4472136  voicing
## 6       p146214 0.6250000 0.5175492    vowel
## 7       p162417 0.5000000 0.5107539   stress
## 8       p162417 0.6000000 0.5070926  voicing
## 9       p162417 0.8750000 0.3535534    vowel
## 10      p188768 0.4736842 0.5129892   stress
## 11      p188768 0.9090909 0.3015113  voicing
## 12      p188768 0.5714286 0.5345225    vowel
## 13       p34258 0.7916667 0.4148511   stress
## 14       p34258 0.6000000 0.5070926  voicing
## 15       p34258 0.7500000 0.4629100    vowel
## 16      p463319 0.4761905 0.5117663   stress
## 17      p463319 0.4666667 0.5163978  voicing
## 18      p463319 0.3333333 0.5163978    vowel
## 19      p466179 0.6250000 0.4945354   stress
## 20      p466179 0.6000000 0.5070926  voicing
## 21      p466179 0.8750000 0.3535534    vowel
## 22      p491113 0.7083333 0.4643056   stress
## 23      p491113 0.9333333 0.2581989  voicing
## 24      p491113 0.8750000 0.3535534    vowel
## 25      p603448 0.4166667 0.5036102   stress
## 26      p603448 0.3750000 0.5000000  voicing
## 27      p603448 0.3750000 0.5175492    vowel
## 28      p731581 0.7500000 0.4423259   stress
## 29      p731581 0.9375000 0.2500000  voicing
## 30      p731581 0.7500000 0.4629100    vowel
## 31      p874028 0.7083333 0.4643056   stress
## 32      p874028 0.8125000 0.4031129  voicing
## 33      p874028 0.6250000 0.5175492    vowel
## 34      p929263 0.7500000 0.4423259   stress
## 35      p929263 0.5333333 0.5163978  voicing
## 36      p929263 0.7500000 0.4629100    vowel
## 37      p960446 0.4545455 0.5096472   stress
## 38      p960446 0.5714286 0.5135526  voicing
## 39      p960446 0.5000000 0.5345225    vowel
## 40      p965925 0.5000000 0.5107539   stress
## 41      p965925 0.6875000 0.4787136  voicing
## 42      p965925 0.3750000 0.5175492    vowel
```

```
## 43     p995032 0.5833333 0.5149287   stress
## 44     p995032 0.4000000 0.5163978  voicing
## 45     p995032 0.8333333 0.4082483    vowel
```

show the above by participant

```
AMbyPar = table %>% filter(correct == "A") %>% group_by(participant) %>% summarise(acc = mean(accuracy)
BMbyPar = table %>% filter(correct == "B") %>% group_by(participant) %>% summarise(acc = mean(accuracy)
meansABbyPar = data.frame(rbind(AMbyPar, BMbyPar))
meansABbyPar$correct = rep(c("A", "B"), each = number.of.participants) #adding a labeling column
meansABbyPar
```

```
##    participant        acc        sd correct
## 1      p123572 0.33333333 0.4830459       A
## 2      p146214 0.66666667 0.4815434       A
## 3      p162417 0.50000000 0.5107539       A
## 4      p188768 0.43750000 0.5123475       A
## 5       p34258 0.56521739 0.5068698       A
## 6      p463319 0.05000000 0.2236068       A
## 7      p466179 0.43478261 0.5068698       A
## 8      p491113 0.75000000 0.4423259       A
## 9      p603448 0.04166667 0.2041241       A
## 10     p731581 0.66666667 0.4815434       A
## 11     p874028 0.54166667 0.5089774       A
## 12     p929263 0.62500000 0.4945354       A
## 13     p960446 0.08695652 0.2881041       A
## 14     p965925 0.29166667 0.4643056       A
## 15     p995032 0.57142857 0.5135526       A
## 16     p123572 0.38095238 0.4976134       B
## 17     p146214 0.66666667 0.4815434       B
## 18     p162417 0.69565217 0.4704720       B
## 19     p188768 0.76190476 0.4364358       B
## 20      p34258 0.87500000 0.3378320       B
## 21     p463319 0.81818182 0.3947710       B
## 22     p466179 0.87500000 0.3378320       B
## 23     p491113 0.86956522 0.3443502       B
## 24     p603448 0.75000000 0.4423259       B
## 25     p731581 0.95833333 0.2041241       B
## 26     p874028 0.91666667 0.2823299       B
## 27     p929263 0.73913043 0.4489778       B
## 28     p960446 0.95238095 0.2182179       B
## 29     p965925 0.79166667 0.4148511       B
## 30     p995032 0.57142857 0.5135526       B
```

The above reflects that p463319, p603448, and p960446 almost always answered B, and thus have very low accuracy for X=A items and high accuracy for X=B items.

remove excluded participants

```
table = subset(table, participant != 'p34258') #exposed to both training conditions
table = subset(table, participant != 'p123572') #too low accuracy on ABX task
table = subset(table, participant != 'p463319') #too low accuracy on ABX task
table = subset(table, participant != 'p960446') #too low accuracy on ABX task
table = subset(table, participant != 'p603448') #too low accuracy on ABX task
head(table,10)
```

```
##    participant training LexTale trialIndex item correct      Xtype contrast    pair response accuracy
```

```
## 1      p929263   stress   71.25          1  i44       A irrelevant  voicing   faces        B        0
## 2      p929263   stress   71.25          2   i5       A    extreme   stress extract        A        1
## 3      p929263   stress   71.25          3  i48       A irrelevant  voicing    dose        B        0
## 4      p929263   stress   71.25          4   i2       B    extreme   stress  refund        B        1
## 5      p929263   stress   71.25          5  i46       A irrelevant  voicing   faces        A        1
## 6      p929263   stress   71.25          6  i47       B irrelevant  voicing    dose        A        0
## 7      p929263   stress   71.25          7  i32       A irrelevant    vowel  litter        A        1
## 8      p929263   stress   71.25          8  i27       A irrelevant    vowel  living        B        0
## 9      p929263   stress   71.25          9  i39       A irrelevant  voicing    dose        B        0
## 10     p929263   stress   71.25         10  i30       A irrelevant    vowel  litter        A        1
```

```
tail(table,10)
```

```
##     participant training LexTale trialIndex item correct      Xtype contrast    pair response accura
## 700     p162417   stress   76.25         38  i26       B irrelevant    vowel    list        B       1
## 701     p162417   stress   76.25         39  i21       A  ambiguous   stress conduct        A       1
## 702     p162417   stress   76.25         40  i17       A  ambiguous   stress extract        A       1
## 703     p162417   stress   76.25         41  i33       A irrelevant  voicing  device        B       1
## 704     p162417   stress   76.25         42  i24       A  ambiguous   stress conduct        B       0
## 705     p162417   stress   76.25         43  i42       B irrelevant  voicing  device        A       0
## 706     p162417   stress   76.25         44  i22       B  ambiguous   stress conduct        B       1
## 707     p162417   stress   76.25         45  i20       A  ambiguous   stress extract        B       0
## 708     p162417   stress   76.25         46  i29       A irrelevant    vowel  living        B       1
## 710     p162417   stress   76.25         48   i8       A    extreme   stress extract        A       1
```

update number of participants

```
number.of.participants = 10 #now excluding 5 participants
```

```
AM = table %>% filter(correct == "A") %>% summarise(acc = mean(accuracy), sd = sd(accuracy))
BM = table %>% filter(correct == "B") %>% summarise(acc = mean(accuracy), sd = sd(accuracy))
meansCor = data.frame(rbind(AM, BM))
meansCor$correct = c("A","B") #adding a labeling column
meansCor$err = 1- meansCor$acc #adding error column
meansCor
```

By if X=A or X=B is correct

```
##         acc        sd correct       err
## 1 0.5520362 0.4984138       A 0.4479638
## 2 0.7946429 0.4048671       B 0.2053571
```

and by item

```
itemAcc = table %>% group_by(item, pair, correct) %>% summarise(acc = mean(accuracy), sd = sd(accuracy))
```

```
## `summarise()` has grouped output by 'item', 'pair'. You can override using the `.groups` argument.
```

```
itemAcc = itemAcc %>% arrange (pair)
itemAcc
```

```
## # A tibble: 48 x 5
## # Groups:   item, pair [48]
##    item  pair    correct   acc    sd
##    <fct> <fct>   <fct>   <dbl> <dbl>
##  1 i10   conduct B       0.8   0.422
##  2 i11   conduct B       0.556 0.527
##  3 i12   conduct A       0.222 0.441
```

```
##  4 i21   conduct A       0.667 0.5
##  5 i22   conduct B       0.6   0.516
##  6 i23   conduct B       0.8   0.422
##  7 i24   conduct A       0.444 0.527
##  8 i9    conduct A       0.75  0.463
##  9 i33   device  A       0.6   0.516
## 10 i34   device  B       0.778 0.441
## # i 38 more rows
```

and by participant

```
AMbyPar = table %>% filter(correct == "A") %>% group_by(participant) %>% summarise(acc = mean(accuracy)
BMbyPar = table %>% filter(correct == "B") %>% group_by(participant) %>% summarise(acc = mean(accuracy)
meansABbyPar = data.frame(rbind(AMbyPar, BMbyPar))
meansABbyPar$correct = rep(c("A", "B"), each = number.of.participants) #adding a labeling column
meansABbyPar %>% arrange(participant)
```

```
##    participant       acc        sd correct
## 1       p146214 0.6666667 0.4815434       A
## 2       p146214 0.6666667 0.4815434       B
## 3       p162417 0.5000000 0.5107539       A
## 4       p162417 0.6956522 0.4704720       B
## 5       p188768 0.4375000 0.5123475       A
## 6       p188768 0.7619048 0.4364358       B
## 7       p466179 0.4347826 0.5068698       A
## 8       p466179 0.8750000 0.3378320       B
## 9       p491113 0.7500000 0.4423259       A
## 10      p491113 0.8695652 0.3443502       B
## 11      p731581 0.6666667 0.4815434       A
## 12      p731581 0.9583333 0.2041241       B
## 13      p874028 0.5416667 0.5089774       A
## 14      p874028 0.9166667 0.2823299       B
## 15      p929263 0.6250000 0.4945354       A
## 16      p929263 0.7391304 0.4489778       B
## 17      p965925 0.2916667 0.4643056       A
## 18      p965925 0.7916667 0.4148511       B
## 19      p995032 0.5714286 0.5135526       A
## 20      p995032 0.5714286 0.5135526       B
```

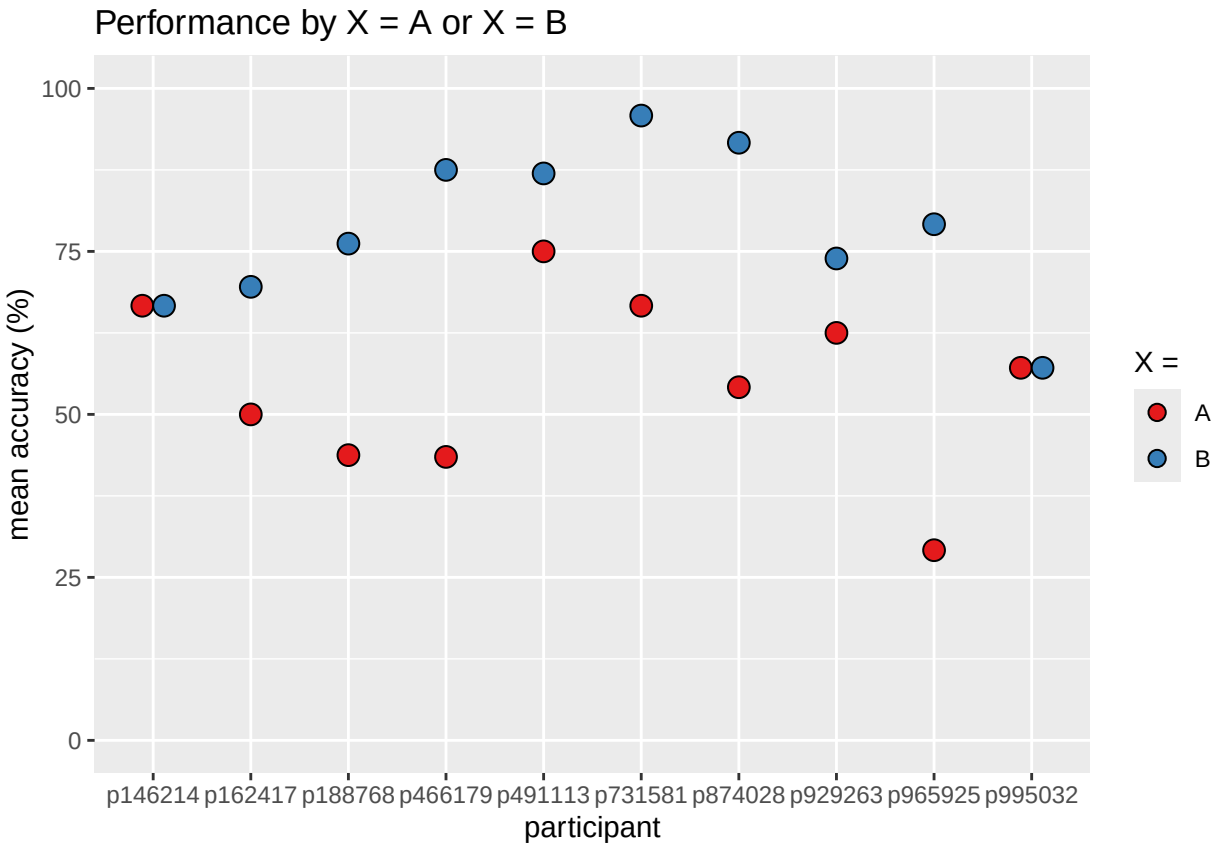
Note: Plot below is Figure 3 in paper.

```
p <- ggplot(data = meansABbyPar, aes(x= participant, fill = correct, y = acc*100))+

    ## defining plot type and y-axis variable

    geom_dotplot(binaxis= "y", stackdir = "center", stackgroups = TRUE) +

    scale_fill_brewer(palette="Set1") +

  ylim(0,100) +

  labs(y= "mean accuracy (%)", title = "Performance by X = A or X = B", fill = "X =")
```

```
## `geom_dotplot()` called with `stackgroups = TRUE` and `method = "dotdensity"`.", i = "Do you want `bi
```

```
p
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with `binwidth`.
```



Prepare to run a model for this:

Set contrasts

```
levels (table$correct)
```

```
## [1] "A" "B"
```

```
corcontrast <- cbind (c(-0.5, 0.5))
colnames (corcontrast) <- c("-A+B")
contrasts (table$correct) <- corcontrast
contrasts (table$correct)
```

```
##   -A+B
## A -0.5
## B  0.5
```

Relevant model:

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```r
corModel = glmer(formula = accuracy ~ correct + (correct | participant) + (1 | item), data = table, fami
summary(corModel)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
##  Family: binomial  ( logit )
## Formula: accuracy ~ correct + (correct | participant) + (1 | item)
##    Data: table
##
##      AIC      BIC   logLik deviance df.resid
##    540.4    565.0   -264.2    528.4      439
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5223 -1.0248  0.4587  0.7494  1.1415
##
## Random effects:
##  Groups      Name        Variance Std.Dev. Corr
##  item        (Intercept) 0.02397  0.1548
##  participant (Intercept) 0.09717  0.3117
##              correct-A+B 0.24620  0.4962   0.23
## Number of obs: 445, groups:  item, 48; participant, 10
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.8021     0.1516   5.292 1.21e-07 ***
## correct-A+B   1.1875     0.2773   4.283 1.85e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## correct-A+B 0.230
```

```r
confidence.intervals <- confint.merMod (corModel, method="Wald")
confidence.intervals
```

```
##                 2.5 %   97.5 %
## .sig01             NA       NA
## .sig02             NA       NA
## .sig03             NA       NA
## .sig04             NA       NA
## (Intercept) 0.5050215 1.099141
## correct-A+B 0.6440678 1.730966
```

```r
exp(1.1875) #main effect of correct response (X=A or B)
```

```
## [1] 3.278874
```

```r
exp(confidence.intervals)
```

```
##                2.5 %   97.5 %
## .sig01            NA       NA
## .sig02            NA       NA
## .sig03            NA       NA
## .sig04            NA       NA
## (Intercept) 1.657021 3.001587
```

```
## correct-A+B 1.904211 5.646104
```

Basic report: For our participants, odds of responding correctly were 3.28 times greater when X matched B than when X matched A (z = 4.283, p from 1 = 1.85 x 10^-5, with a confidence interval running from 1.90 to 5.65). [then generalize to population]

Were participants more accurate when they responded A or when they responded B?

```
AMbyPar = table %>% filter (response == "A") %>% group_by(participant) %>% summarise(acc = mean(accuracy
BMbyPar = table %>% filter(response == "B") %>% group_by(participant) %>% summarise(acc = mean(accuracy)
meansABbyPar = data.frame(rbind(AMbyPar, BMbyPar))
meansABbyPar$response = rep(c("A", "B"), each = number.of.participants) #adding a labeling column
meansABbyPar %>% arrange(participant)
```

```
##    participant       acc        sd response
## 1       p146214 0.6666667 0.4815434        A
## 2       p146214 0.6666667 0.4815434        B
## 3       p162417 0.6315789 0.4955946        A
## 4       p162417 0.5714286 0.5039526        B
## 5       p188768 0.5833333 0.5149287        A
## 6       p188768 0.6400000 0.4898979        B
## 7       p466179 0.7692308 0.4385290        A
## 8       p466179 0.6176471 0.4932702        B
## 9       p491113 0.8571429 0.3585686        A
## 10      p491113 0.7692308 0.4296689        B
## 11      p731581 0.9411765 0.2425356        A
## 12      p731581 0.7419355 0.4448027        B
## 13      p874028 0.8666667 0.3518658        A
## 14      p874028 0.6666667 0.4787136        B
## 15      p929263 0.7142857 0.4629100        A
## 16      p929263 0.6538462 0.4851645        B
## 17      p965925 0.5833333 0.5149287        A
## 18      p965925 0.5277778 0.5063094        B
## 19      p995032 0.5714286 0.5135526        A
## 20      p995032 0.5714286 0.5135526        B
```

```
library(ggplot2)
p <- ggplot(data = meansABbyPar, aes(x= participant, fill = response, y = acc*100))+

    ## defining plot type and y-axis variable

    geom_dotplot(binaxis= "y", stackdir = "center", stackgroups = TRUE) +

    scale_fill_brewer(palette="Set1") +

  ylim(0,100)
```

```
## `geom_dotplot()` called with `stackgroups = TRUE` and `method = "dotdensity"`.", i = "Do you want `bi
```

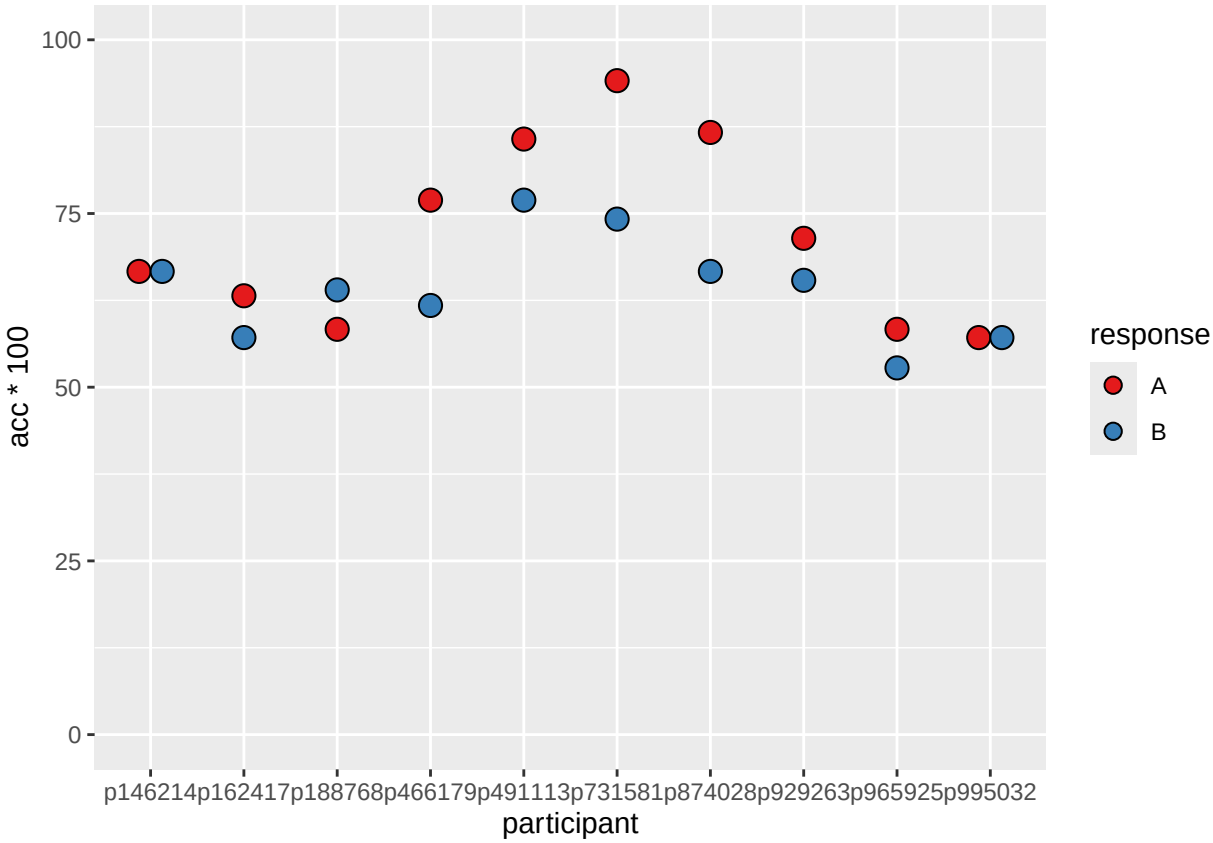
```
  labs(y= "mean accuracy (%)", title = "Performance by X = A or X = B", fill = "X =")
```

```
## $y
## [1] "mean accuracy (%)"
##
## $fill
## [1] "X ="
##
```

```
## $title
## [1] "Performance by X = A or X = B"
##
## attr(,"class")
## [1] "labels"
```

```
p
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with `binwidth`.
```



Analyze accuracy by minimal pair

now do calculations for the mean and standard deviation for each pair

```
refM = table %>% filter(pair == "refund") %>% summarise(acc = mean(accuracy), sd = sd(accuracy))
extM = table %>% filter(pair == "extract") %>% summarise(acc = mean(accuracy), sd = sd(accuracy))
conM = table %>% filter(pair == "conduct") %>% summarise(acc = mean(accuracy), sd = sd(accuracy))
devM = table %>% filter(pair == "device") %>% summarise(acc = mean(accuracy), sd = sd(accuracy))
facM = table %>% filter(pair == "faces") %>% summarise(acc = mean(accuracy), sd = sd(accuracy))
dosM = table %>% filter(pair == "dose") %>% summarise(acc = mean(accuracy), sd = sd(accuracy))
meansP = data.frame(rbind(refM, extM, conM, devM, facM, dosM))
meansP$pair = c("refund", "extract", "conduct", "device", "faces", "dose") #adding a labeling column
meansP$err = 1- meansP$acc #adding error column
meansP
```

```
##         acc        sd    pair       err
## 1 0.6753247 0.4713240  refund 0.3246753
## 2 0.5972222 0.4938986 extract 0.4027778
```

```
## 3 0.6081081 0.4915050 conduct 0.3918919
## 4 0.7555556 0.4346135  device 0.2444444
## 5 0.7115385 0.4574670   faces 0.2884615
## 6 0.7083333 0.4593396    dose 0.2916667
```

Note: this seems to show, and implications for contrast coding: - refund most accurate (+.5, -1/3) - conduct in the middle (0, +2/3) - extract least accurate (-.5, -1/3) contrast 1: +R-E contrast 2: +C-RE

Analysis by contrast type

do calculations for the mean and standard deviation for each contrast type

```
strM = table %>% filter(contrast == "stress") %>% summarise(acc = mean(accuracy), sd = sd(accuracy))
voiM = table %>% filter(contrast == "voicing") %>% summarise(acc = mean(accuracy), sd = sd(accuracy))
vowM = table %>% filter(contrast == "vowel") %>% summarise(acc = mean(accuracy), sd = sd(accuracy))
meansC = data.frame(rbind(strM, voiM, vowM))
meansC$contrast = c("stress","voicing", "vowel") #adding a labeling column
meansC$err = 1- meansC$acc #adding error column
meansC
```

```
##         acc        sd contrast       err
## 1 0.6278027 0.4844781   stress 0.3721973
## 2 0.7241379 0.4484969  voicing 0.2758621
## 3 0.7142857 0.4547163    vowel 0.2857143
```

By contrast type and training

```
strM = table %>% group_by(training) %>% filter(contrast == "stress") %>% summarise(acc = mean(accuracy)
voiM = table %>% group_by(training) %>% filter(contrast == "voicing") %>% summarise(acc = mean(accuracy
vowM = table %>% group_by(training) %>% filter(contrast == "vowel") %>% summarise(acc = mean(accuracy),
meansC = data.frame(rbind(strM, voiM, vowM))
meansC$contrast = rep(c("stress","voicing", "vowel"), each =2) #adding a labeling column, 2 for number
meansC$err = 1- meansC$acc #adding error column
meansC
```

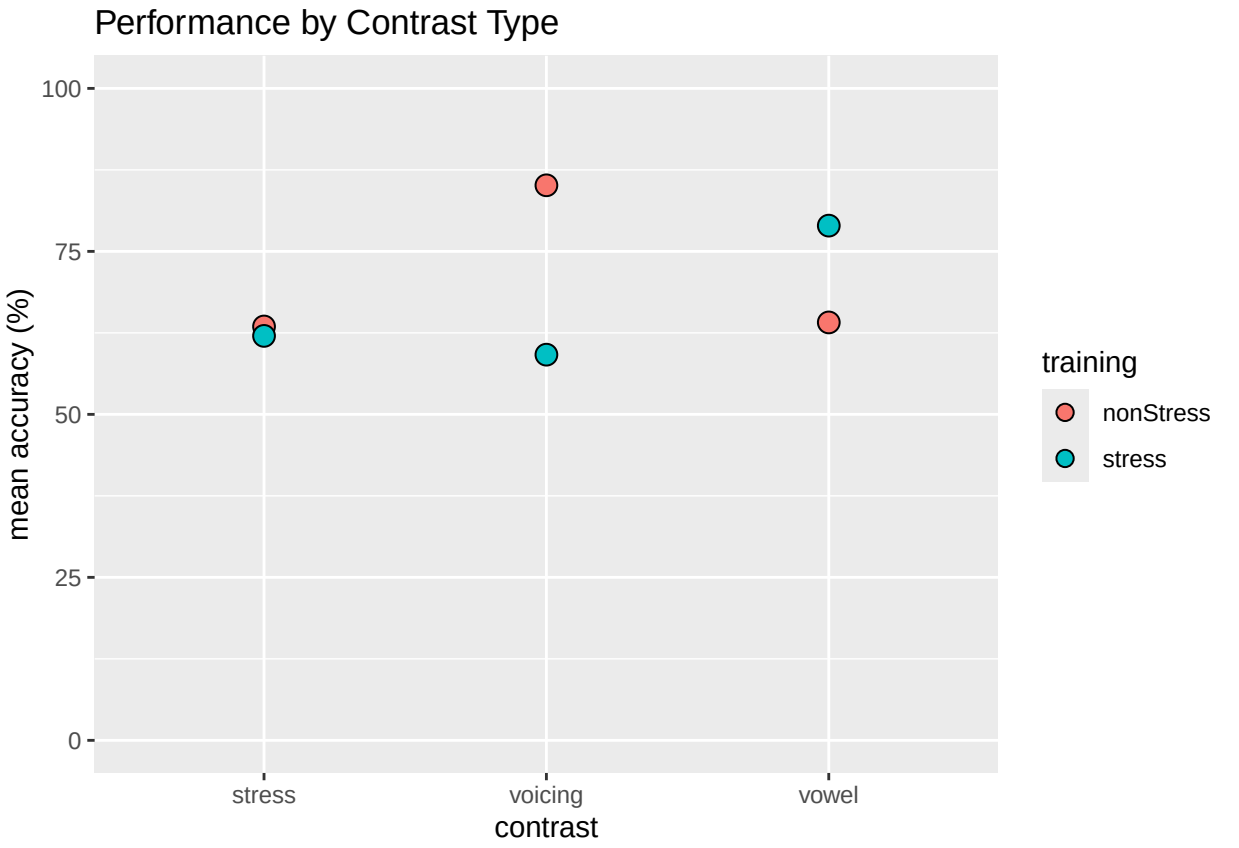
```
##    training       acc        sd contrast       err
## 1 nonStress 0.6347826 0.4835983   stress 0.3652174
## 2    stress 0.6203704 0.4875572   stress 0.3796296
## 3 nonStress 0.8513514 0.3581701  voicing 0.1486486
## 4    stress 0.5915493 0.4950459  voicing 0.4084507
## 5 nonStress 0.6410256 0.4859705    vowel 0.3589744
## 6    stress 0.7894737 0.4131550    vowel 0.2105263
```

```
library(ggplot2)
p <- ggplot(data = meansC, aes(x= contrast, fill = training, y = acc*100))+

    ## defining plot type and y-axis variable

    geom_dotplot(binaxis= "y", stackdir = "center", stackgroups = TRUE) +

  labs(y= "mean accuracy (%)", title = "Performance by Contrast Type") +

  ylim (0, 100)
```

```
## `geom_dotplot()` called with `stackgroups = TRUE` and `method = "dotdensity"`.", i = "Do you want `bi
```

```
p
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with `binwidth`.
```



do calculations for the mean and standard deviation for each contrast type by participant

```
strM = table %>% group_by(participant) %>% filter(contrast == "stress") %>% summarise(acc = mean(accura
voiM = table %>% group_by(participant) %>% filter(contrast == "voicing") %>% summarise(acc = mean(accur
vowM = table %>% group_by(participant) %>% filter(contrast == "vowel") %>% summarise(acc = mean(accurac
meansCbyPar = data.frame(rbind(strM, voiM, vowM))
meansCbyPar$contrast = rep(c("stress","voicing", "vowel"), each = number.of.participants) #adding a lab

# Adding training column
meansCbyPar =
meansCbyPar %>%
  mutate(training = case_when(
    (participant == "p929263") ~ "Stress",
   (participant == "p466179") ~ "Stress",
    (participant == "p146214") ~ "Stress",
     (participant == "p995032") ~ "Stress",
   (participant == "p162417") ~ "Stress",
    (participant == "p965925") ~ "nonStress",
    (participant == "p731581") ~ "nonStress",
    (participant == "p874028") ~ "nonStress",
   (participant == "p491113") ~ "nonStress",
    (participant == "p188768") ~ "nonStress"))
meansCbyPar %>% arrange(participant)
```

```
##    participant       acc        sd contrast  training
## 1       p146214 0.6250000 0.4945354   stress    Stress
## 2       p146214 0.7500000 0.4472136  voicing    Stress
## 3       p146214 0.6250000 0.5175492    vowel    Stress
## 4       p162417 0.5000000 0.5107539   stress    Stress
## 5       p162417 0.6000000 0.5070926  voicing    Stress
## 6       p162417 0.8750000 0.3535534    vowel    Stress
## 7       p188768 0.4736842 0.5129892   stress nonStress
## 8       p188768 0.9090909 0.3015113  voicing nonStress
## 9       p188768 0.5714286 0.5345225    vowel nonStress
## 10      p466179 0.6250000 0.4945354   stress    Stress
## 11      p466179 0.6000000 0.5070926  voicing    Stress
## 12      p466179 0.8750000 0.3535534    vowel    Stress
## 13      p491113 0.7083333 0.4643056   stress nonStress
## 14      p491113 0.9333333 0.2581989  voicing nonStress
## 15      p491113 0.8750000 0.3535534    vowel nonStress
## 16      p731581 0.7500000 0.4423259   stress nonStress
## 17      p731581 0.9375000 0.2500000  voicing nonStress
## 18      p731581 0.7500000 0.4629100    vowel nonStress
## 19      p874028 0.7083333 0.4643056   stress nonStress
## 20      p874028 0.8125000 0.4031129  voicing nonStress
## 21      p874028 0.6250000 0.5175492    vowel nonStress
## 22      p929263 0.7500000 0.4423259   stress    Stress
## 23      p929263 0.5333333 0.5163978  voicing    Stress
## 24      p929263 0.7500000 0.4629100    vowel    Stress
## 25      p965925 0.5000000 0.5107539   stress nonStress
## 26      p965925 0.6875000 0.4787136  voicing nonStress
## 27      p965925 0.3750000 0.5175492    vowel nonStress
## 28      p995032 0.5833333 0.5149287   stress    Stress
## 29      p995032 0.4000000 0.5163978  voicing    Stress
## 30      p995032 0.8333333 0.4082483    vowel    Stress
```

```
meansCbyPar <- meansCbyPar %>% group_by(contrast) %>% mutate(upper =  quantile(acc, 0.75),
                                                             lower = quantile(acc, 0.25),
                                                             mean = mean(acc)*100)
meansCbyPar
```

```
## # A tibble: 30 x 8
## # Groups:   contrast [3]
##    participant   acc    sd contrast training  upper lower  mean
##    <fct>       <dbl> <dbl> <chr>    <chr>     <dbl> <dbl> <dbl>
##  1 p146214     0.625 0.495 stress   Stress    0.708 0.521  62.2
##  2 p162417     0.5   0.511 stress   Stress    0.708 0.521  62.2
##  3 p188768     0.474 0.513 stress   nonStress 0.708 0.521  62.2
##  4 p466179     0.625 0.495 stress   Stress    0.708 0.521  62.2
##  5 p491113     0.708 0.464 stress   nonStress 0.708 0.521  62.2
##  6 p731581     0.75  0.442 stress   nonStress 0.708 0.521  62.2
##  7 p874028     0.708 0.464 stress   nonStress 0.708 0.521  62.2
##  8 p929263     0.75  0.442 stress   Stress    0.708 0.521  62.2
##  9 p965925     0.5   0.511 stress   nonStress 0.708 0.521  62.2
## 10 p995032     0.583 0.515 stress   Stress    0.708 0.521  62.2
## # i 20 more rows
```

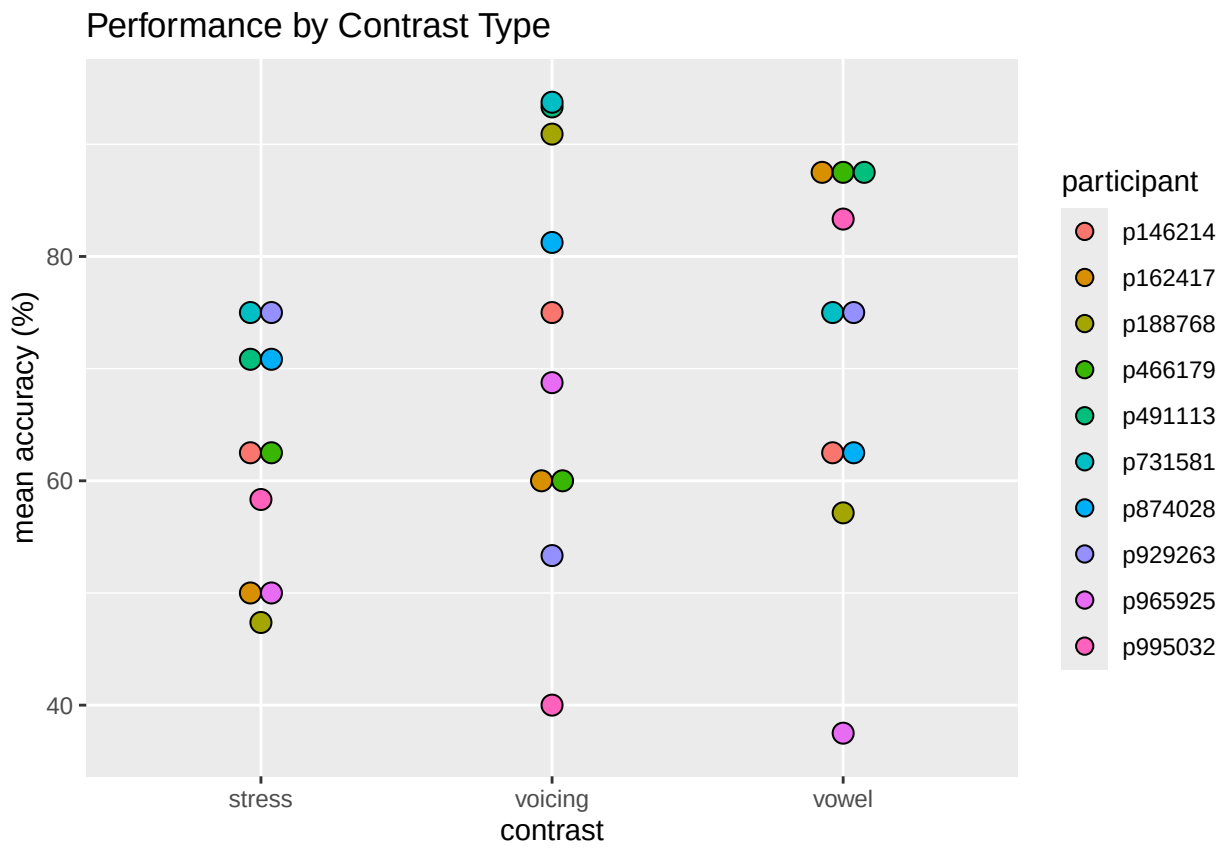
I want to plot the mean performance by contrast type per participant to show the individual variation, and that ceiling is highest for voicing contrast participant: discrete accuracy: y, continuous contrast: discrete

```
p <- ggplot(data = meansCbyPar, aes(x= contrast, fill = participant, y = acc*100))+

    ## defining plot type and y-axis variable

    geom_dotplot(binaxis= "y", stackdir = "center", stackgroups = TRUE) +

  labs(y= "mean accuracy (%)", title = "Performance by Contrast Type")
```

```
## `geom_dotplot()` called with `stackgroups = TRUE` and `method = "dotdensity"`.", i = "Do you want `bi
```

```
p
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with `binwidth`.
```



same but by training

Note: Plot below is Figure 2 in paper.

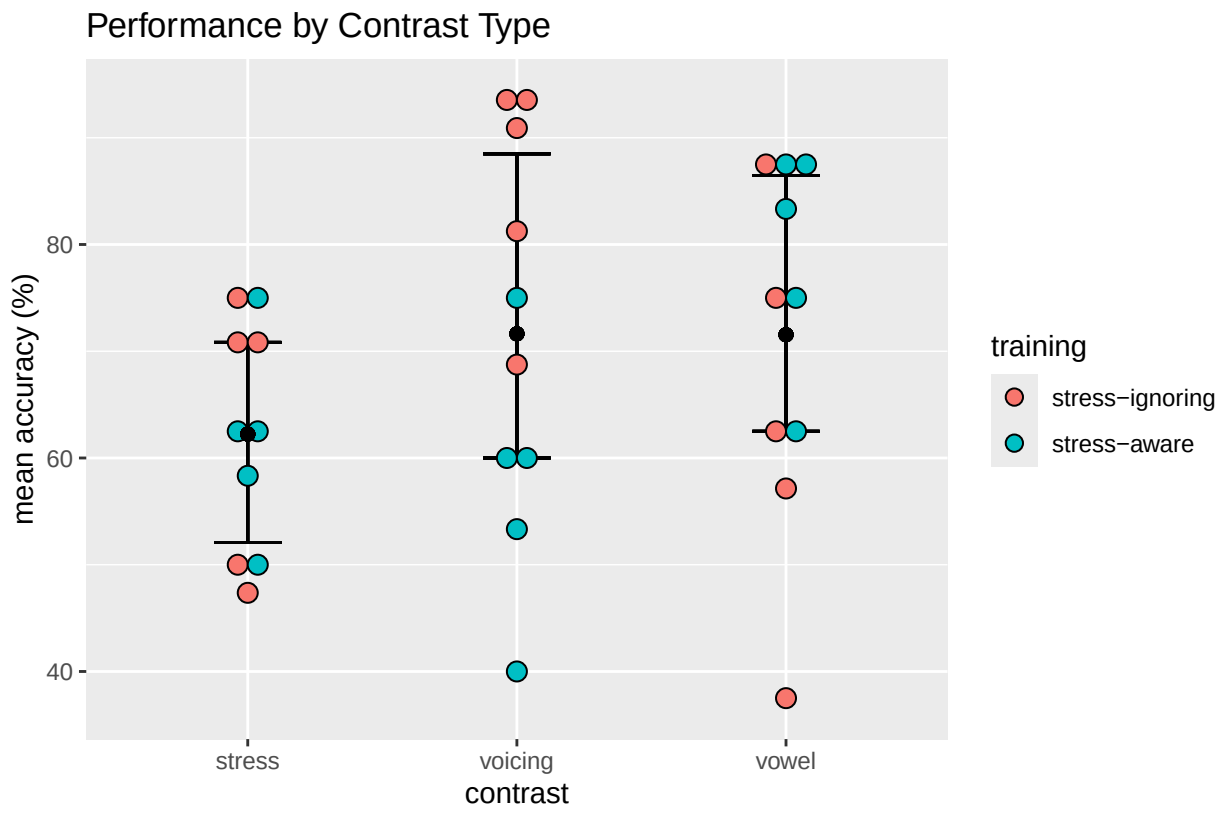
```
library(ggplot2)
p <- ggplot(data = meansCbyPar, aes(x= contrast, fill = training, y = acc*100))+

    ## defining plot type and y-axis variable

  geom_errorbar(aes(ymin = lower*100, ymax = upper*100),col = "black",
                width =  0.25) +

  geom_dotplot(binaxis= "y", stackdir = "center", stackgroups = TRUE) +

  geom_point(aes(x = contrast, y = mean), size = 2, col = "black", show.legend = FALSE) +
```

```
  scale_fill_discrete(labels = c("stress-ignoring", "stress-aware")) +

  labs(y= "mean accuracy (%)", title = "Performance by Contrast Type", caption = "Note: Each colored do
```

```
## `geom_dotplot()` called with `stackgroups = TRUE` and `method = "dotdensity"`.", i = "Do you want `bi
```

```
p
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with `binwidth`.
```



ote: Each colored dot represents one participant's average accuracy on the given contrast.

By minimal pair and by participant

do calculations for the mean and standard deviation for each minimal pair by participant

```
library(tidyverse)
library(scrutiny)
refM = table %>% group_by(participant) %>% filter(pair == "refund") %>% summarise(acc = mean(accuracy),
extM = table %>% group_by(participant) %>% filter(pair == "extract") %>% summarise(acc = mean(accuracy)
conM = table %>% group_by(participant) %>% filter(pair == "conduct") %>% summarise(acc = mean(accuracy)
devM = table %>% group_by(participant) %>% filter(pair == "device") %>% summarise(acc = mean(accuracy),
facM = table %>% group_by(participant) %>% filter(pair == "faces") %>% summarise(acc = mean(accuracy), s
dosM = table %>% group_by(participant) %>% filter(pair == "dose") %>% summarise(acc = mean(accuracy), sc
meansPbyPar = data.frame(rbind(refM, extM, conM, devM, facM, dosM))
meansPbyPar$pair = rep(c("refund", "extract", "conduct", "device", "faces", "dose"), each = number.of.pa
meansPbyPar$err = 1- meansPbyPar$acc #adding error column
meansPbyPar
```

```
##    participant       acc        sd    pair       err
## 1      p146214 0.7500000 0.4629100  refund 0.2500000
## 2      p162417 0.3750000 0.5175492  refund 0.6250000
## 3      p188768 0.4285714 0.5345225  refund 0.5714286
## 4      p466179 0.6250000 0.5175492  refund 0.3750000
## 5      p491113 0.8750000 0.3535534  refund 0.1250000
## 6      p731581 0.7500000 0.4629100  refund 0.2500000
## 7      p874028 0.7500000 0.4629100  refund 0.2500000
## 8      p929263 0.8750000 0.3535534  refund 0.1250000
## 9      p965925 0.5000000 0.5345225  refund 0.5000000
## 10     p995032 0.8333333 0.4082483  refund 0.1666667
## 11     p146214 0.5000000 0.5345225 extract 0.5000000
## 12     p162417 0.5000000 0.5345225 extract 0.5000000
## 13     p188768 0.6666667 0.5163978 extract 0.3333333
## 14     p466179 0.5000000 0.5345225 extract 0.5000000
## 15     p491113 0.7500000 0.4629100 extract 0.2500000
## 16     p731581 0.6250000 0.5175492 extract 0.3750000
## 17     p874028 0.7500000 0.4629100 extract 0.2500000
## 18     p929263 0.7500000 0.4629100 extract 0.2500000
## 19     p965925 0.5000000 0.5345225 extract 0.5000000
## 20     p995032 0.0000000 0.0000000 extract 1.0000000
## 21     p146214 0.6250000 0.5175492 conduct 0.3750000
## 22     p162417 0.6250000 0.5175492 conduct 0.3750000
## 23     p188768 0.3333333 0.5163978 conduct 0.6666667
## 24     p466179 0.7500000 0.4629100 conduct 0.2500000
## 25     p491113 0.5000000 0.5345225 conduct 0.5000000
## 26     p731581 0.8750000 0.3535534 conduct 0.1250000
## 27     p874028 0.6250000 0.5175492 conduct 0.3750000
## 28     p929263 0.6250000 0.5175492 conduct 0.3750000
## 29     p965925 0.5000000 0.5345225 conduct 0.5000000
## 30     p995032 0.5000000 0.5773503 conduct 0.5000000
## 31     p146214 0.8000000 0.4472136  device 0.2000000
## 32     p162417 0.6000000 0.5477226  device 0.4000000
## 33     p188768 1.0000000 0.0000000  device 0.0000000
## 34     p466179 0.4000000 0.5477226  device 0.6000000
## 35     p491113 1.0000000 0.0000000  device 0.0000000
## 36     p731581 1.0000000 0.0000000  device 0.0000000
## 37     p874028 0.8000000 0.4472136  device 0.2000000
## 38     p929263 0.8000000 0.4472136  device 0.2000000
## 39     p965925 0.8000000 0.4472136  device 0.2000000
## 40     p995032 0.3333333 0.5773503  device 0.6666667
## 41     p146214 0.6666667 0.5163978   faces 0.3333333
## 42     p162417 0.5000000 0.5477226   faces 0.5000000
## 43     p188768 1.0000000 0.0000000   faces 0.0000000
## 44     p466179 0.6000000 0.5477226   faces 0.4000000
## 45     p491113 0.8333333 0.4082483   faces 0.1666667
## 46     p731581 0.8333333 0.4082483   faces 0.1666667
## 47     p874028 1.0000000 0.0000000   faces 0.0000000
## 48     p929263 0.6000000 0.5477226   faces 0.4000000
## 49     p965925 0.6666667 0.5163978   faces 0.3333333
## 50     p995032 0.3333333 0.5773503   faces 0.6666667
## 51     p146214 0.8000000 0.4472136    dose 0.2000000
## 52     p162417 0.7500000 0.5000000    dose 0.2500000
## 53     p188768 0.8000000 0.4472136    dose 0.2000000
```

```
## 54     p466179 0.8000000 0.4472136    dose 0.2000000
## 55     p491113 1.0000000 0.0000000    dose 0.0000000
## 56     p731581 1.0000000 0.0000000    dose 0.0000000
## 57     p874028 0.6000000 0.5477226    dose 0.4000000
## 58     p929263 0.2000000 0.4472136    dose 0.8000000
## 59     p965925 0.6000000 0.5477226    dose 0.4000000
## 60     p995032 0.5000000 0.5773503    dose 0.5000000
```

```
# same thing but only for stress minimal pairs
meansPbyStressPar = data.frame(rbind(refM, extM, conM))
meansPbyStressPar$pair = rep(c("refund", "extract", "conduct"),  each = number.of.participants) #adding
meansPbyStressPar$err = 1- meansPbyStressPar$acc #adding error column
meansPbyStressPar %>% arrange(participant)
```

```
##    participant       acc        sd    pair       err
## 1      p146214 0.7500000 0.4629100  refund 0.2500000
## 2      p146214 0.5000000 0.5345225 extract 0.5000000
## 3      p146214 0.6250000 0.5175492 conduct 0.3750000
## 4      p162417 0.3750000 0.5175492  refund 0.6250000
## 5      p162417 0.5000000 0.5345225 extract 0.5000000
## 6      p162417 0.6250000 0.5175492 conduct 0.3750000
## 7      p188768 0.4285714 0.5345225  refund 0.5714286
## 8      p188768 0.6666667 0.5163978 extract 0.3333333
## 9      p188768 0.3333333 0.5163978 conduct 0.6666667
## 10     p466179 0.6250000 0.5175492  refund 0.3750000
## 11     p466179 0.5000000 0.5345225 extract 0.5000000
## 12     p466179 0.7500000 0.4629100 conduct 0.2500000
## 13     p491113 0.8750000 0.3535534  refund 0.1250000
## 14     p491113 0.7500000 0.4629100 extract 0.2500000
## 15     p491113 0.5000000 0.5345225 conduct 0.5000000
## 16     p731581 0.7500000 0.4629100  refund 0.2500000
## 17     p731581 0.6250000 0.5175492 extract 0.3750000
## 18     p731581 0.8750000 0.3535534 conduct 0.1250000
## 19     p874028 0.7500000 0.4629100  refund 0.2500000
## 20     p874028 0.7500000 0.4629100 extract 0.2500000
## 21     p874028 0.6250000 0.5175492 conduct 0.3750000
## 22     p929263 0.8750000 0.3535534  refund 0.1250000
## 23     p929263 0.7500000 0.4629100 extract 0.2500000
## 24     p929263 0.6250000 0.5175492 conduct 0.3750000
## 25     p965925 0.5000000 0.5345225  refund 0.5000000
## 26     p965925 0.5000000 0.5345225 extract 0.5000000
## 27     p965925 0.5000000 0.5345225 conduct 0.5000000
## 28     p995032 0.8333333 0.4082483  refund 0.1666667
## 29     p995032 0.0000000 0.0000000 extract 1.0000000
## 30     p995032 0.5000000 0.5773503 conduct 0.5000000
```

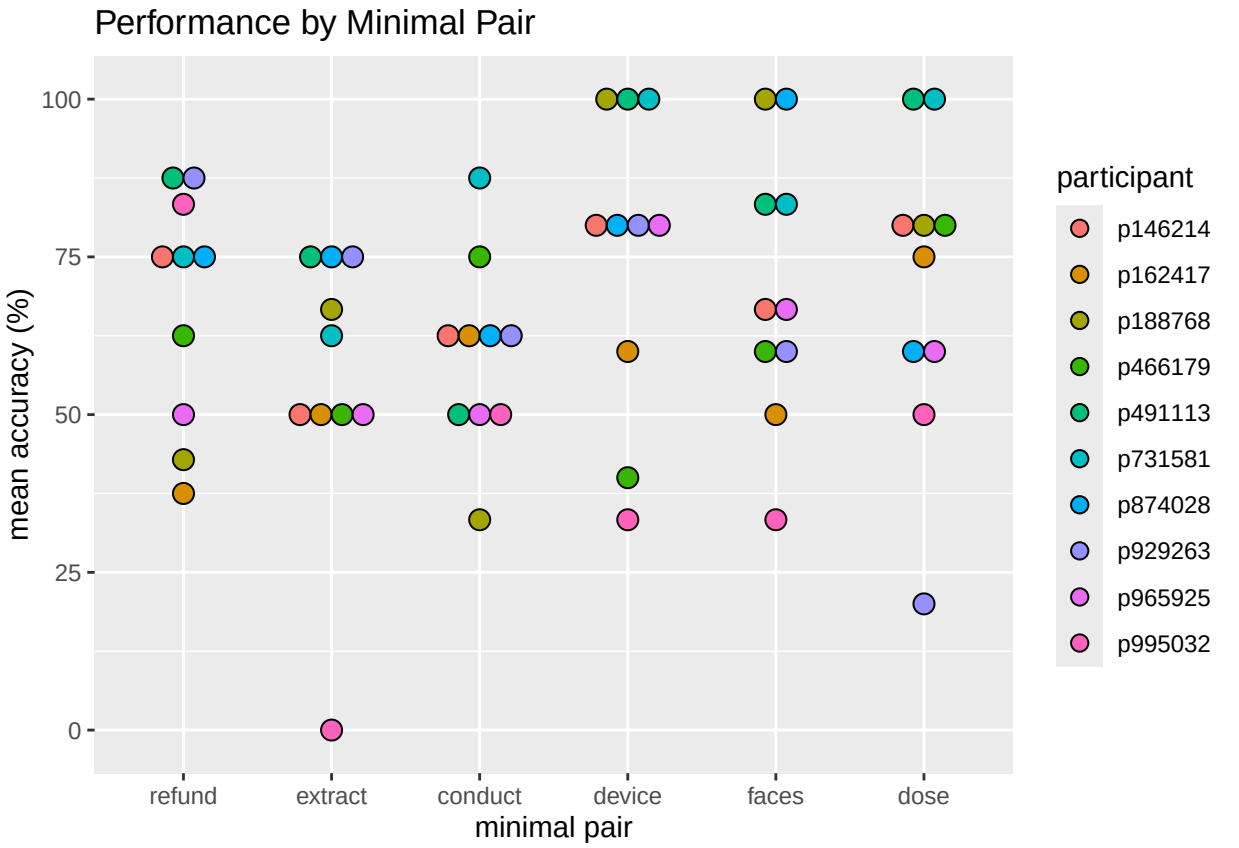
```
library(ggplot2)

ordered_pairs = c("refund", "extract", "conduct", "device", "faces", "dose")

b <- ggplot(data = meansPbyPar, aes(x = factor (pair, ordered_pairs), fill = participant, y = acc*100))

    ## defining plot type and y-axis variable

    geom_dotplot(binaxis= "y", stackdir = "center", stackgroups = TRUE) +
```

```
  labs(y= "mean accuracy (%)",  x = "minimal pair", title = "Performance by Minimal Pair")
```

```
## `geom_dotplot()` called with `stackgroups = TRUE` and `method = "dotdensity"`.", i = "Do you want `bi
```

```
b
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with `binwidth`.
```



means by training and contrast, also subdivided by pair Below is information needed to construct Table 2 in draft

```
refM = table %>% group_by(training) %>% filter(pair == "refund") %>% summarise(acc = mean(accuracy), sd
extM = table %>% group_by(training) %>% filter(pair == "extract") %>% summarise(acc = mean(accuracy), s
conM = table %>% group_by(training) %>% filter(pair == "conduct") %>% summarise(acc = mean(accuracy), s
devM = table %>% group_by(training) %>% filter(pair == "device") %>% summarise(acc = mean(accuracy), sd
facM = table %>% group_by(training) %>% filter(pair == "faces") %>% summarise(acc = mean(accuracy), sd =
dosM = table %>% group_by(training) %>% filter(pair == "dose") %>% summarise(acc = mean(accuracy), sd =
StConMbytr = table %>% group_by(training) %>% filter(contrast == "stress") %>% summarise(acc = mean(accu
VoConMbytr = table %>% group_by(training) %>% filter(contrast == "voicing") %>% summarise(acc = mean(ac
meansbycontrastandtraining = data.frame(rbind(StConMbytr, VoConMbytr))
meansbycontrastandtraining$contrast = rep(c("stress", "voicing"), each = 2) #adding a labeling column
meansbycontrastandtraining$err = 1- meansbycontrastandtraining$acc #adding error column
meansbycontrastandtraining
```

```
##    training       acc        sd contrast       err
## 1 nonStress 0.6347826 0.4835983   stress 0.3652174
## 2    stress 0.6203704 0.4875572   stress 0.3796296
## 3 nonStress 0.8513514 0.3581701  voicing 0.1486486
```

```
## 4    stress 0.5915493 0.4950459  voicing 0.4084507
```

```
meansPbyTr = data.frame(rbind(refM, extM, conM, devM, facM, dosM))
meansPbyTr$pair = rep(c("refund", "extract", "conduct", "device", "faces", "dose"), each = 2) #adding a
meansPbyTr$err = 1- meansPbyTr$acc #adding error column
meansPbyTr %>% arrange (training)
```

```
##     training       acc        sd    pair        err
## 1  nonStress 0.6666667 0.4775669  refund 0.33333333
## 2  nonStress 0.6578947 0.4807829 extract 0.34210526
## 3  nonStress 0.5789474 0.5003555 conduct 0.42105263
## 4  nonStress 0.9090909 0.2942449  device 0.09090909
## 5  nonStress 0.8518519 0.3620140   faces 0.14814815
## 6  nonStress 0.8000000 0.4082483    dose 0.20000000
## 7     stress 0.6842105 0.4710691  refund 0.31578947
## 8     stress 0.5294118 0.5066404 extract 0.47058824
## 9     stress 0.6388889 0.4871361 conduct 0.36111111
## 10    stress 0.6086957 0.4990109  device 0.39130435
## 11    stress 0.5600000 0.5066228   faces 0.44000000
## 12    stress 0.6086957 0.4990109    dose 0.39130435
```

```
meansPbyTr
```

```
##     training       acc        sd    pair        err
## 1  nonStress 0.6666667 0.4775669  refund 0.33333333
## 2     stress 0.6842105 0.4710691  refund 0.31578947
## 3  nonStress 0.6578947 0.4807829 extract 0.34210526
## 4     stress 0.5294118 0.5066404 extract 0.47058824
## 5  nonStress 0.5789474 0.5003555 conduct 0.42105263
## 6     stress 0.6388889 0.4871361 conduct 0.36111111
## 7  nonStress 0.9090909 0.2942449  device 0.09090909
## 8     stress 0.6086957 0.4990109  device 0.39130435
## 9  nonStress 0.8518519 0.3620140   faces 0.14814815
## 10    stress 0.5600000 0.5066228   faces 0.44000000
## 11 nonStress 0.8000000 0.4082483    dose 0.20000000
## 12    stress 0.6086957 0.4990109    dose 0.39130435
```

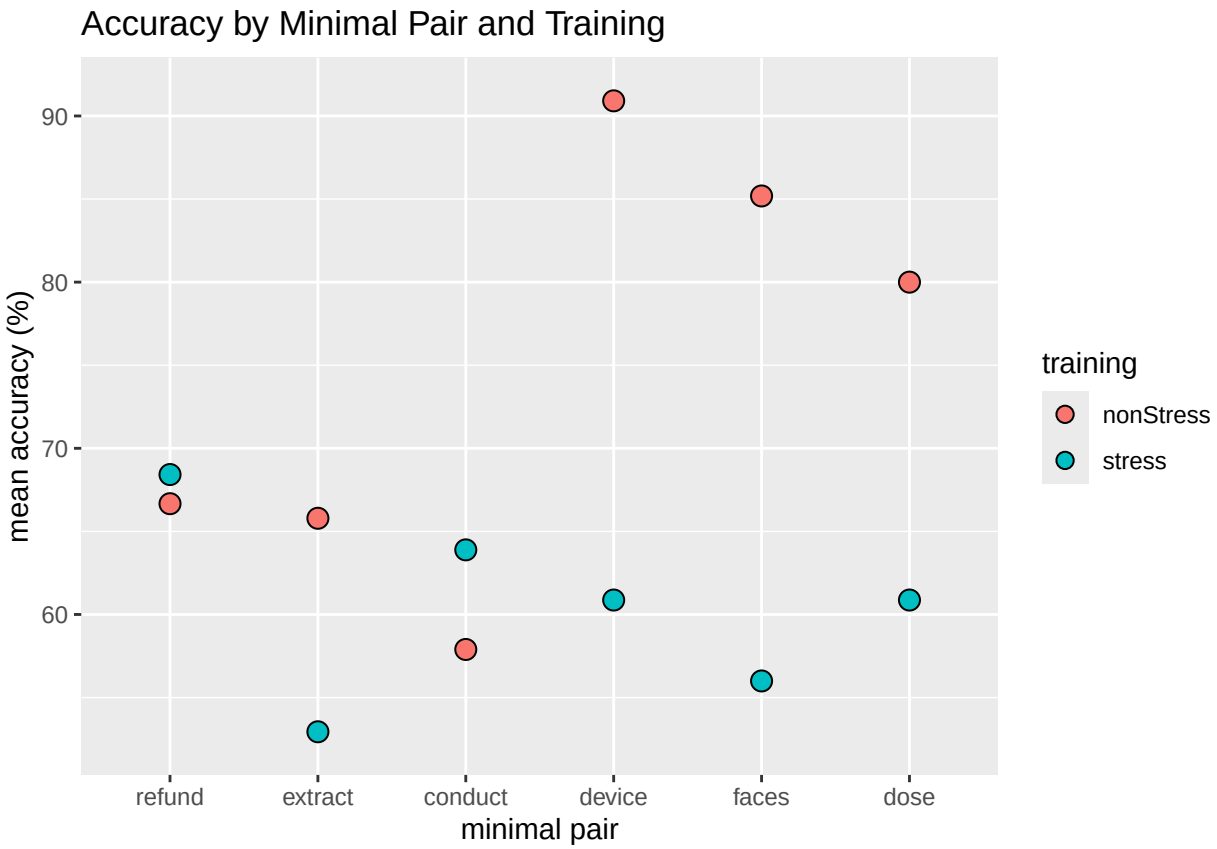
```
# same thing but only for stress minimal pairs
meansPbyStressTr = data.frame(rbind(refM, extM, conM))
meansPbyStressTr$pair = rep(c("refund", "extract", "conduct"), each = 2) #adding a labeling column
meansPbyStressTr$err = 1- meansPbyStressTr$acc #adding error column
```

```
library(ggplot2)
p <- ggplot(data = meansPbyTr, aes(x = factor (pair, ordered_pairs), fill = training, y = acc*100))+

    ## defining plot type and y-axis variable

    geom_dotplot(binaxis= "y", stackdir = "center", stackgroups = TRUE) +

  labs(y= "mean accuracy (%)", x = "minimal pair", title = "Accuracy by Minimal Pair and Training")
```

```
## `geom_dotplot()` called with `stackgroups = TRUE` and `method = "dotdensity"`.", i = "Do you want `bi
```

```
p
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with `binwidth`.
```



I want to see the above graph, but with dots per participant, colored by training

Taking data by participant and contrast, plus adding a column for training

```
library(tidyverse)
refM = table %>% group_by(participant) %>% filter(pair == "refund") %>% summarise(acc = mean(accuracy),
extM = table %>% group_by(participant) %>% filter(pair == "extract") %>% summarise(acc = mean(accuracy)
conM = table %>% group_by(participant) %>% filter(pair == "conduct") %>% summarise(acc = mean(accuracy)
devM = table %>% group_by(participant) %>% filter(pair == "device") %>% summarise(acc = mean(accuracy),
facM = table %>% group_by(participant) %>% filter(pair == "faces") %>% summarise(acc = mean(accuracy), s
dosM = table %>% group_by(participant) %>% filter(pair == "dose") %>% summarise(acc = mean(accuracy), sc
livM = table %>% group_by(participant) %>% filter(pair == "living") %>% summarise(acc = mean(accuracy),
lisM = table %>% group_by(participant) %>% filter(pair == "list") %>% summarise(acc = mean(accuracy), sc
litM = table %>% group_by(participant) %>% filter(pair == "litter") %>% summarise(acc = mean(accuracy),
meansPbyPar = data.frame(rbind(refM, extM, conM, devM, facM, dosM, livM, lisM, litM))
meansPbyPar$pair = rep(c("refund", "extract", "conduct", "device", "faces", "dose", "living", "list", "l
meansPbyPar$err = 1- meansPbyPar$acc #adding error column

# Adding training column
meansPbyPar =
meansPbyPar %>%
  mutate(training = case_when(
    (participant == "p929263") ~ "Stress",
   (participant == "p466179") ~ "Stress",
    (participant == "p146214") ~ "Stress",
     (participant == "p995032") ~ "Stress",
    (participant == "p162417") ~ "Stress",
    (participant == "p965925") ~ "nonStress",
```

```
    (participant == "p731581") ~ "nonStress",
    (participant == "p874028") ~ "nonStress",
   (participant == "p491113") ~ "nonStress",
    (participant == "p188768") ~ "nonStress"))

meansPbyPar <- meansPbyPar %>% group_by(pair) %>% mutate(upper =  quantile(acc, 0.75),
                                                         lower = quantile(acc, 0.25),
                                                         mean = mean(acc)*100)
meansPbyPar
```

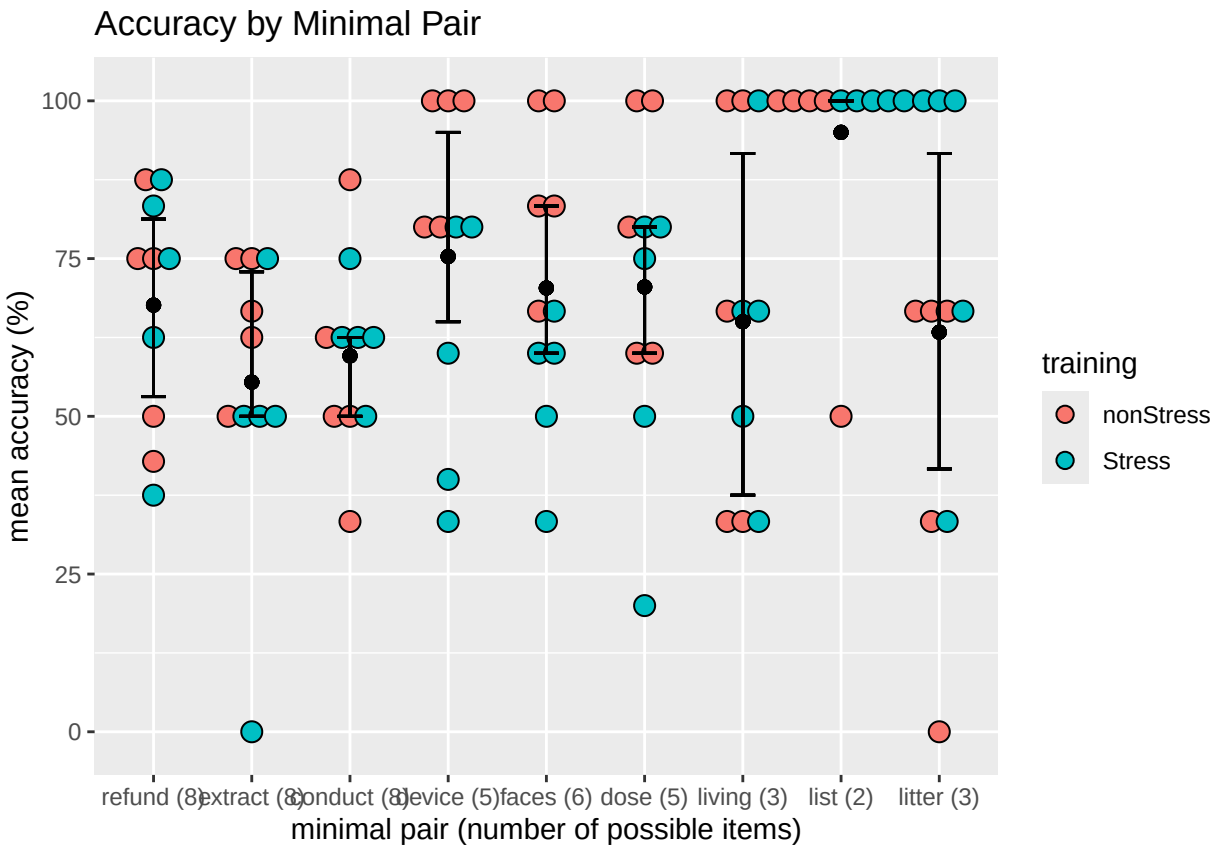
```
## # A tibble: 90 x 9
## # Groups:   pair [9]
##    participant   acc    sd pair     err training  upper lower  mean
##    <fct>       <dbl> <dbl> <chr>  <dbl> <chr>     <dbl> <dbl> <dbl>
##  1 p146214     0.75  0.463 refund 0.25  Stress    0.812 0.531  67.6
##  2 p162417     0.375 0.518 refund 0.625 Stress    0.812 0.531  67.6
##  3 p188768     0.429 0.535 refund 0.571 nonStress 0.812 0.531  67.6
##  4 p466179     0.625 0.518 refund 0.375 Stress    0.812 0.531  67.6
##  5 p491113     0.875 0.354 refund 0.125 nonStress 0.812 0.531  67.6
##  6 p731581     0.75  0.463 refund 0.25  nonStress 0.812 0.531  67.6
##  7 p874028     0.75  0.463 refund 0.25  nonStress 0.812 0.531  67.6
##  8 p929263     0.875 0.354 refund 0.125 Stress    0.812 0.531  67.6
##  9 p965925     0.5   0.535 refund 0.5   nonStress 0.812 0.531  67.6
## 10 p995032     0.833 0.408 refund 0.167 Stress    0.812 0.531  67.6
## # i 80 more rows
```

```
ordered_pairs_long = c("refund", "extract", "conduct", "device", "faces", "dose", "living", "list", "li

p <- ggplot(data = meansPbyPar, aes(x = factor (pair, ordered_pairs_long), fill = training, y = acc*100)

    ## defining plot type and y-axis variable

  geom_dotplot(binaxis= "y", stackdir = "center", stackratio = 0.75, stackgroups = TRUE) +

  geom_errorbar(aes(ymin = lower*100, ymax = upper*100),col = "black", width =  0.25) +

  geom_point(aes(x = pair, y = mean), size = 2, col = "black", show.legend = FALSE) +

  labs(y= "mean accuracy (%)", x = "minimal pair (number of possible items)", title = "Accuracy by Minir

  scale_x_discrete(labels=c("refund (8)", "extract (8)", "conduct (8)", "device (5)", "faces (6)", "dos
```

```
## `geom_dotplot()` called with `stackgroups = TRUE` and `method = "dotdensity"`.", i = "Do you want `bi
```

```
p
```

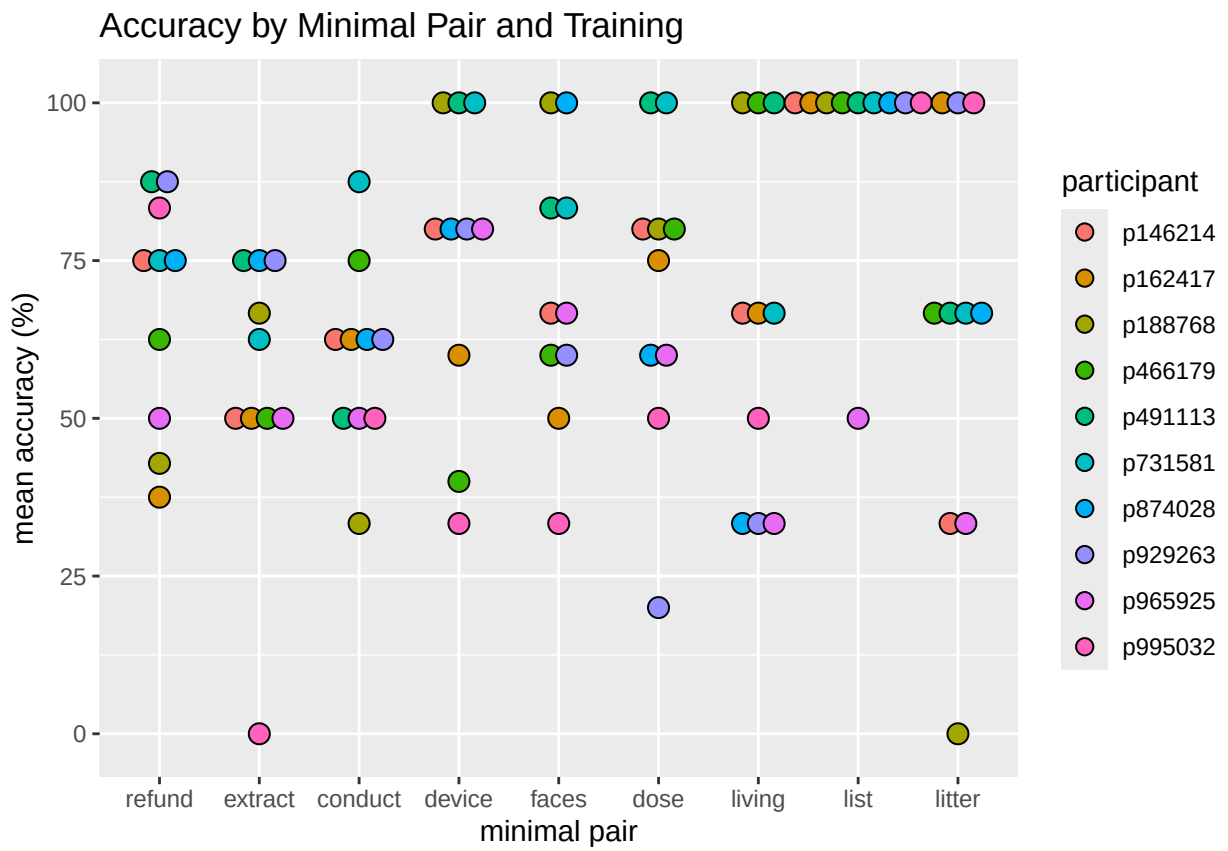
```
## Bin width defaults to 1/30 of the range of the data. Pick better value with `binwidth`.
```




```
ordered_pairs_long = c("refund", "extract", "conduct", "device", "faces", "dose", "living", "list", "li

p <- ggplot(data = meansPbyPar, aes(x = factor (pair, ordered_pairs_long), fill = participant, y = acc*

    ## defining plot type and y-axis variable

    geom_dotplot(binaxis= "y", stackdir = "center", stackratio = 0.75, stackgroups = TRUE) +

  labs(y= "mean accuracy (%)", x = "minimal pair", title = "Accuracy by Minimal Pair and Training")
```

```
## `geom_dotplot()` called with `stackgroups = TRUE` and `method = "dotdensity"`.", i = "Do you want `bi
p
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with `binwidth`.
```



prep to analyze if accuracy for voicing is significantly better than for stress

This would confirm prior research/ phenomenon of stress deafness

take out rows concerning vowel (fewer items, will not consider in this analysis)

```
tableSan = subset(table, contrast != 'vowel') #i.e. for confirmation. This is a sanity check
#tableSan
head(tableSan,10)
```

```
##    participant training LexTale trialIndex item correct      Xtype contrast    pair response accuracy
## 1      p929263   stress   71.25          1  i44       A irrelevant  voicing   faces        B        0
## 2      p929263   stress   71.25          2   i5       A    extreme   stress extract        A        1
## 3      p929263   stress   71.25          3  i48       A irrelevant  voicing    dose        B        0
## 4      p929263   stress   71.25          4   i2       B    extreme   stress  refund        B        1
## 5      p929263   stress   71.25          5  i46       A irrelevant  voicing   faces        A        1
## 6      p929263   stress   71.25          6  i47       B irrelevant  voicing    dose        A        0
## 9      p929263   stress   71.25          9  i39       A irrelevant  voicing    dose        B        0
## 12     p929263   stress   71.25         12  i33       A irrelevant  voicing  device        A        1
## 14     p929263   stress   71.25         14  i19       B  ambiguous   stress extract        B        1
## 15     p929263   stress   71.25         15  i21       A  ambiguous   stress conduct        A        1
```

```
tail(tableSan,10)
```

```
##     participant training LexTale trialIndex item correct      Xtype contrast    pair response accura
## 698     p162417   stress   76.25         36  i10       B    extreme   stress conduct        B
## 699     p162417   stress   76.25         37   i1       A    extreme   stress  refund        B
## 701     p162417   stress   76.25         39  i21       A  ambiguous   stress conduct        A
```

```
## 702     p162417   stress   76.25         40  i17       A  ambiguous   stress extract        A
## 703     p162417   stress   76.25         41  i33       A irrelevant  voicing  device        B
## 704     p162417   stress   76.25         42  i24       A  ambiguous   stress conduct        B
## 705     p162417   stress   76.25         43  i42       B irrelevant  voicing  device        A
## 706     p162417   stress   76.25         44  i22       B  ambiguous   stress conduct        B
## 707     p162417   stress   76.25         45  i20       A  ambiguous   stress extract        B
## 710     p162417   stress   76.25         48   i8       A    extreme   stress extract        A
```

```
write.table(tableSan, file="data/SDsanitycheck.txt", quote = FALSE, row.names = FALSE, sep = "\t")
```

prep to only analyze critical (stress) trials

remove all filler (non-stress contrast) trials

```
table = subset(table, Xtype != 'irrelevant')
#table
head(table,10)
```

```
##    participant training LexTale trialIndex item correct     Xtype contrast    pair response accuracy
## 2      p929263   stress   71.25          2   i5       A   extreme   stress extract        A        1
## 4      p929263   stress   71.25          4   i2       B   extreme   stress  refund        B        1
## 14     p929263   stress   71.25         14  i19       B ambiguous   stress extract        B        1
## 15     p929263   stress   71.25         15  i21       A ambiguous   stress conduct        A        1
## 16     p929263   stress   71.25         16  i18       B ambiguous   stress extract        B        1
## 17     p929263   stress   71.25         17  i14       B ambiguous   stress  refund        B        1
## 18     p929263   stress   71.25         18  i22       B ambiguous   stress conduct        B        1
## 19     p929263   stress   71.25         19  i13       A ambiguous   stress  refund        A        1
## 22     p929263   stress   71.25         22   i4       A   extreme   stress  refund        A        1
## 24     p929263   stress   71.25         24  i17       A ambiguous   stress extract        A        1
```

```
tail(table,10)
```

```
##     participant training LexTale trialIndex item correct     Xtype contrast    pair response accuracy
## 693     p162417   stress   76.25         31   i6       B   extreme   stress extract        B        1
## 696     p162417   stress   76.25         34  i12       A   extreme   stress conduct        B        0
## 698     p162417   stress   76.25         36  i10       B   extreme   stress conduct        B        1
## 699     p162417   stress   76.25         37   i1       A   extreme   stress  refund        B        0
## 701     p162417   stress   76.25         39  i21       A ambiguous   stress conduct        A        1
## 702     p162417   stress   76.25         40  i17       A ambiguous   stress extract        A        1
## 704     p162417   stress   76.25         42  i24       A ambiguous   stress conduct        B        0
## 706     p162417   stress   76.25         44  i22       B ambiguous   stress conduct        B        1
## 707     p162417   stress   76.25         45  i20       A ambiguous   stress extract        B        0
## 710     p162417   stress   76.25         48   i8       A   extreme   stress extract        A        1
```

```
levels(table$participant)
```

```
##  [1] "p123572" "p146214" "p162417" "p188768" "p34258"  "p463319" "p466179" "p491113" "p603448" "p7315
```

The data is now readable with only what I need. Now I read out the data into a new file for analysis

```
write.table(table, file="data/StressConCleaned.txt", quote = FALSE, row.names = FALSE, sep = "\t")
```