SDdataAnalysis

2025-05-30

This file contains the main (RQ-answering) analysis and some secondary analyses based only on the stress-contrast items.

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

```
##   participant training LexTale trialIndex item correct     Xtype contrast
## 1     p929263   stress   71.25          2   i5       A   extreme   stress
## 2     p929263   stress   71.25          4   i2       B   extreme   stress
## 3     p929263   stress   71.25         14  i19       B ambiguous   stress
## 4     p929263   stress   71.25         15  i21       A ambiguous   stress
## 5     p929263   stress   71.25         16  i18       B ambiguous   stress
## 6     p929263   stress   71.25         17  i14       B ambiguous   stress
##      pair response accuracy
## 1 extract        A        1
## 2  refund        B        1
## 3 extract        B        1
## 4 conduct        A        1
## 5 extract        B        1
## 6  refund        B        1
```

```
tail(table)
```

```
##     participant training LexTale trialIndex item correct     Xtype contrast
## 218     p162417   stress   76.25         39  i21       A ambiguous   stress
## 219     p162417   stress   76.25         40  i17       A ambiguous   stress
## 220     p162417   stress   76.25         42  i24       A ambiguous   stress
## 221     p162417   stress   76.25         44  i22       B ambiguous   stress
## 222     p162417   stress   76.25         45  i20       A ambiguous   stress
## 223     p162417   stress   76.25         48   i8       A   extreme   stress
##        pair response accuracy
## 218 conduct        A        1
## 219 extract        A        1
## 220 conduct        B        0
## 221 conduct        B        1
## 222 extract        B        0
## 223 extract        A        1
```

make a little plot for performance by minimal pair in stress contrast to initially visualize relevant data

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## 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 ---------------------------------------- tidyverse_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
```

```
stressM = table %>% group_by(participant) %>% filter(training == "stress") %>% summarise(acc = mean(accu
nonstressM = table %>% group_by(participant) %>% filter(training == "nonStress") %>% summarise(acc = mea
meansTrbyPar = data.frame(rbind(stressM, nonstressM))
meansTrbyPar = meansTrbyPar %>%
  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"))
meansTrbyPar
```

```
##    participant       acc        sd  training
## 1       p146214 0.6250000 0.4945354    Stress
## 2       p162417 0.5000000 0.5107539    Stress
## 3       p466179 0.6250000 0.4945354    Stress
## 4       p929263 0.7500000 0.4423259    Stress
## 5       p995032 0.5833333 0.5149287    Stress
## 6       p188768 0.4736842 0.5129892 nonStress
## 7       p491113 0.7083333 0.4643056 nonStress
## 8       p731581 0.7500000 0.4423259 nonStress
## 9       p874028 0.7083333 0.4643056 nonStress
## 10      p965925 0.5000000 0.5107539 nonStress
```

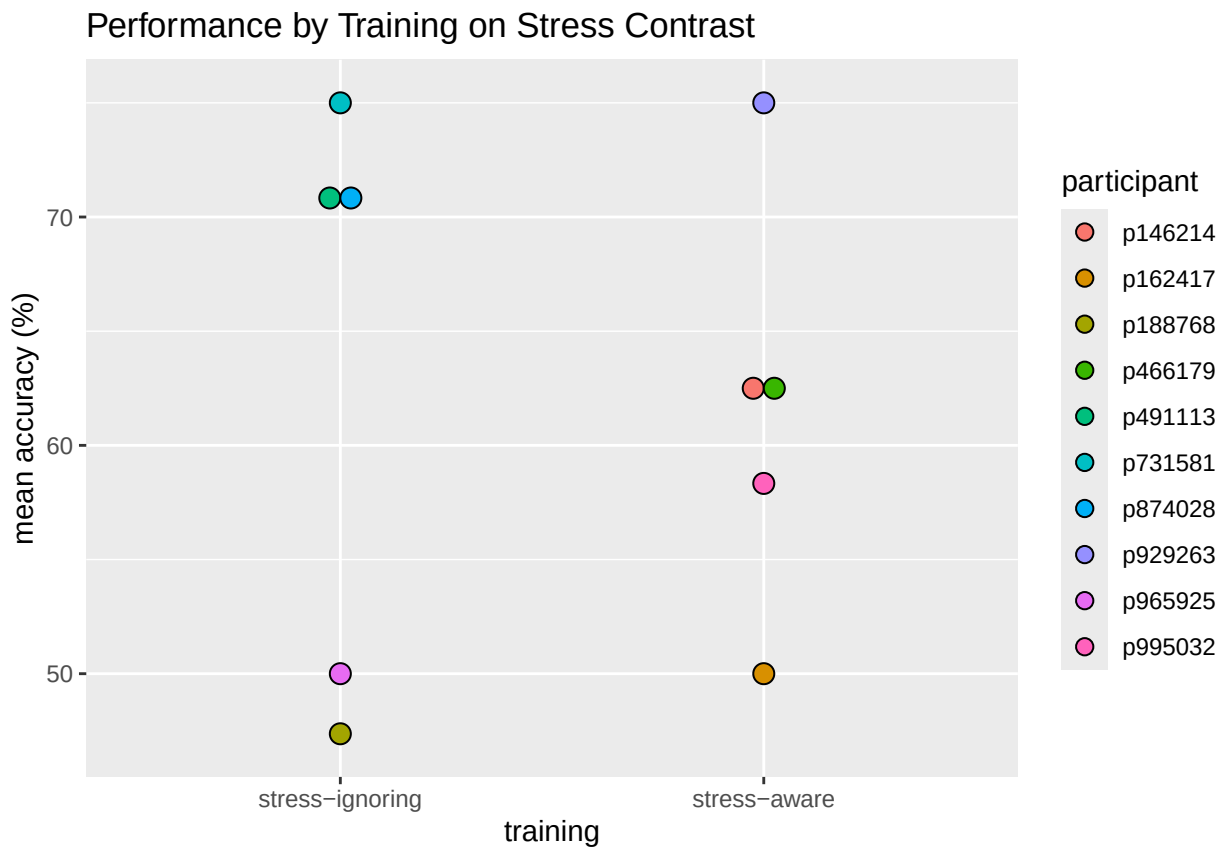
Note: Plot below is Figure 1 in paper

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

    ## defining plot type and y-axis variable

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

  scale_x_discrete(label = c("stress-ignoring", "stress-aware")) +

  labs(y= "mean accuracy (%)", title = "Performance by Training on Stress Contrast")
```

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

```
p
```

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



set contrasts for

- training
- Xtype
- (minimal) pair
- correct (if it's supposed to be A or B)

```
levels (table$training)
```

```
## [1] "nonStress" "stress"
```

```
traincontrast <- cbind (c(-0.5, 0.5))
colnames (traincontrast) <- c("-nonStress+stress")
contrasts (table$training) <- traincontrast
contrasts (table$training)
```

```
##           -nonStress+stress
## nonStress              -0.5
## stress                  0.5
```

```
levels (table$Xtype)
```

```
## [1] "ambiguous" "extreme"
```

```
Xcontrast <- cbind (c(-0.5, 0.5))
colnames (Xcontrast) <- c("-amb+ext")
contrasts (table$Xtype) <- Xcontrast
contrasts (table$Xtype)
```

```
##           -amb+ext
```

```
## ambiguous     -0.5
## extreme        0.5
```

```
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
```

```
library(lme4)
```

RQ-answering minimal model

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

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

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

```
minModel = glmer(formula = accuracy ~ training + (1 | participant) + (training | item), data = table, fa
summary(minModel)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: accuracy ~ training + (1 | participant) + (training | item)
##    Data: table
##
##      AIC      BIC   logLik deviance df.resid
##    300.4    320.9   -144.2    288.4      217
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8036 -1.0353  0.5133  0.7621  1.1428
##
## Random effects:
##  Groups      Name                        Variance Std.Dev. Corr
##  item        (Intercept)                 0.39887  0.6316
##              training-nonStress+stress 0.85463  0.9245   -1.00
##  participant (Intercept)                 0.04483  0.2117
## Number of obs: 223, groups:  item, 24; participant, 10
##
## Fixed effects:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  0.5877     0.2104   2.794  0.00521 **
## training-nonStress+stress  -0.1802     0.3819  -0.472  0.63699
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## trnng-nnSt+ -0.375
```

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

```
##                                2.5 %    97.5 %
## .sig01                            NA        NA
## .sig02                            NA        NA
## .sig03                            NA        NA
## .sig04                            NA        NA
## (Intercept)                0.1753641 1.0000166
## training-nonStress+stress -0.9287282 0.5682862
```

```
exp(0.1802) #main effect of training
```

```
## [1] 1.197457
```

```
exp (-confidence.intervals)
```

```
##                               2.5 %    97.5 %
## .sig01                           NA        NA
## .sig02                           NA        NA
## .sig03                           NA        NA
## .sig04                           NA        NA
## (Intercept)               0.8391515 0.3678733
## training-nonStress+stress 2.5312879 0.5664955
```

```
exp(.928)
```

```
## [1] 2.529445
```

```
exp(-.568)
```

```
## [1] 0.5666576
```

Odds of responding correctly to stress-contrast trials were 1.197 times greater for our participants who followed the stress-ignoring lesson than those who followed the stress-aware lesson, but not significantly so (z = .472, p from 1 = .637, with a confidence interval running from .566 to 2.531). We thus are unable to conclude if a lesson effects stress perception in French L1 speakers and cannot answer our research question.

```
library(lme4)
corModel = glmer(formula = accuracy ~ correct + (correct | participant) + (1 | item), data = table, fami
```

model for effect of X=A or X=B but only in Stress contrast items

```
## boundary (singular) fit: see help('isSingular')
```

```
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
##    282.7    303.1   -135.3    270.7      217
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5656 -0.9056  0.5092  0.6698  1.8073
##
## Random effects:
##  Groups      Name        Variance Std.Dev. Corr
##  item        (Intercept) 0.00000  0.0000
##  participant (Intercept) 0.02744  0.1656
##              correct-A+B 1.35853  1.1656   -0.85
## Number of obs: 223, groups:  item, 24; participant, 10
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.5812     0.1634   3.556 0.000376 ***
## correct-A+B   1.2384     0.4827   2.566 0.010293 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## correct-A+B -0.142
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```
library(lme4)
trainingcorModel = glmer(formula = accuracy ~ training * correct + ( 1 | participant) + (training | ite
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(trainingcorModel)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: accuracy ~ training * correct + (1 | participant) + (training |
##     item)
##    Data: table
##
##      AIC      BIC   logLik deviance df.resid
##    280.4    307.7   -132.2    264.4      215
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7548 -0.8364  0.3725  0.7759  1.3868
##
## Random effects:
##  Groups      Name                        Variance Std.Dev.  Corr
##  item        (Intercept)                 0.00e+00 0.000e+00
##              training-nonStress+stress 4.44e-10 2.107e-05  NaN
##  participant (Intercept)                 6.55e-02 2.559e-01
## Number of obs: 223, groups:  item, 24; participant, 10
##
```

```
## Fixed effects:
##                                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)                           0.6021     0.1750   3.440 0.000581 ***
## training-nonStress+stress            -0.2143     0.3498  -0.613 0.540143
## correct-A+B                           1.3170     0.3137   4.198  2.7e-05 ***
## training-nonStress+stress:correct-A+B -2.0314     0.6246  -3.252 0.001145 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) trn-S+ cr-A+B
## trnng-nnSt+ -0.122
## correct-A+B  0.169 -0.152
## trn-S+:-A+B -0.159  0.163 -0.182
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

Fit is singular, so X=A or X=B being "correct" cannot be used to improve the minimal model

X Type analyses

```
library(lme4)
XtypeModel = glmer(formula = accuracy ~ Xtype + (Xtype | participant) + (1 | item), data = table, family
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(XtypeModel)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: accuracy ~ Xtype + (Xtype | participant) + (1 | item)
##    Data: table
##
##      AIC      BIC   logLik deviance df.resid
##    300.5    320.9   -144.2    288.5      217
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8517 -1.0346  0.5614  0.7481  1.1020
##
## Random effects:
##  Groups      Name          Variance Std.Dev. Corr
##  item        (Intercept)   0.2414   0.4913
##  participant (Intercept)   0.1261   0.3552
##              Xtype-amb+ext 0.8759   0.9359   -1.00
## Number of obs: 223, groups:  item, 24; participant, 10
##
## Fixed effects:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.5771     0.2132   2.706  0.00681 **
## Xtype-amb+ext  -0.1991     0.4662  -0.427  0.66926
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Correlation of Fixed Effects:
##             (Intr)
## Xtype-mb+xt -0.371
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

remove random slopes to see if fit is still singular

```
library(lme4)
XtypeModel = glmer(formula = accuracy ~ Xtype + (1 | participant) + (1 | item), data = table, family=bi
summary(XtypeModel)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: accuracy ~ Xtype + (1 | participant) + (1 | item)
##    Data: table
##
##      AIC      BIC   logLik deviance df.resid
##    300.9    314.5   -146.5    292.9      219
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.5690 -1.1597  0.6643  0.7422  0.9949
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  item        (Intercept) 0.17404  0.4172
##  participant (Intercept) 0.01426  0.1194
## Number of obs: 223, groups:  item, 24; participant, 10
##
## Fixed effects:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.5426     0.1711   3.172  0.00151 **
## Xtype-amb+ext  -0.1633     0.3309  -0.493  0.62170
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr)
## Xtype-mb+xt -0.005
```

not singular but I think not valid anymore

add a plot for effect of ambiguous vs. prototypical (extreme) X token

```
library(tidyverse)
XtypeM = table %>% group_by(Xtype) %>% summarise(acc = mean(accuracy), sd = sd(accuracy))
meansXtype = data.frame(XtypeM)
meansXtype
```

```
##       Xtype       acc        sd
## 1 ambiguous 0.6460177 0.4803338
## 2   extreme 0.6090909 0.4901873
```

```
refM = table %>% group_by(Xtype) %>% filter(pair == "refund") %>% summarise(acc = mean(accuracy), sd = 
extM = table %>% group_by(Xtype) %>% filter(pair == "extract") %>% summarise(acc = mean(accuracy), sd =
```

```
conM = table %>% group_by(Xtype) %>% filter(pair == "conduct") %>% summarise(acc = mean(accuracy), sd =
meansXtype = data.frame(rbind(refM, extM, conM))
meansXtype$pair = rep(c("refund", "extract", "conduct"), each = 2) #adding a labeling column, each is f
meansXtype$err = 1- meansXtype$acc #adding error column
meansXtype
```

```
##       Xtype       acc        sd    pair       err
## 1 ambiguous 0.7105263 0.4596059  refund 0.2894737
## 2   extreme 0.6410256 0.4859705  refund 0.3589744
## 3 ambiguous 0.5945946 0.4977427 extract 0.4054054
## 4   extreme 0.6000000 0.4970501 extract 0.4000000
## 5 ambiguous 0.6315789 0.4888515 conduct 0.3684211
## 6   extreme 0.5833333 0.5000000 conduct 0.4166667
```

```
library(tidyverse)
XtypeE = table %>% group_by(item) %>% filter(Xtype == "extreme") %>% summarise(acc = mean(accuracy), sd
XtypeA = table %>% group_by(item) %>% filter(Xtype == "ambiguous") %>% summarise(acc = mean(accuracy), s
meansXtype2 = data.frame(rbind(XtypeE, XtypeA))
meansXtype2$Xtype = rep(c("extreme", "ambiguous"), each = 12) #adding a labeling column, each is for nu
meansXtype2
```

```
##    item       acc        sd     Xtype
## 1    i1 0.4000000 0.5163978   extreme
## 2   i10 0.8000000 0.4216370   extreme
## 3   i11 0.5555556 0.5270463   extreme
## 4   i12 0.2222222 0.4409586   extreme
## 5    i2 0.7000000 0.4830459   extreme
## 6    i3 0.9000000 0.3162278   extreme
## 7    i4 0.5555556 0.5270463   extreme
## 8    i5 0.4444444 0.5270463   extreme
## 9    i6 0.7500000 0.4629100   extreme
## 10   i7 0.7777778 0.4409586   extreme
## 11   i8 0.4444444 0.5270463   extreme
## 12   i9 0.7500000 0.4629100   extreme
## 13  i13 0.5555556 0.5270463 ambiguous
## 14  i14 0.8000000 0.4216370 ambiguous
## 15  i15 0.8888889 0.3333333 ambiguous
## 16  i16 0.6000000 0.5163978 ambiguous
## 17  i17 0.3333333 0.5000000 ambiguous
## 18  i18 0.8888889 0.3333333 ambiguous
## 19  i19 0.7000000 0.4830459 ambiguous
## 20  i20 0.4444444 0.5270463 ambiguous
## 21  i21 0.6666667 0.5000000 ambiguous
## 22  i22 0.6000000 0.5163978 ambiguous
## 23  i23 0.8000000 0.4216370 ambiguous
## 24  i24 0.4444444 0.5270463 ambiguous
```

```
library(tidyverse)
refM = table %>% group_by(Xtype, item, .add = TRUE) %>% filter(pair == "refund") %>% summarise(acc = mea
```

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

```
extM = table %>% group_by(Xtype, item, .add = TRUE) %>% filter(pair == "extract") %>% summarise(acc = me
```

```
## `summarise()` has grouped output by 'Xtype'. You can override using the
```

```
## `.groups` argument.
```

```
conM = table %>% group_by(Xtype, item, .add = TRUE) %>% filter(pair == "conduct") %>% summarise(acc = m
```

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

```
meansXtype = data.frame(rbind(refM, extM, conM))
meansXtype$pair = rep(c("refund", "extract", "conduct"), each = 2) #adding a labeling column, each is f
meansXtype$err = 1- meansXtype$acc #adding error column
meansXtype <- meansXtype %>% group_by(pair, Xtype, .add = TRUE) %>% mutate(upper =  quantile(acc, 0.75)
                                                    lower = quantile(acc, 0.25),
                                                    mean = mean(acc)*100)
meansXtype %>% arrange(pair)
```

```
## # A tibble: 24 x 9
## # Groups:   pair, Xtype [6]
##    Xtype     item    acc    sd pair      err upper lower  mean
##    <fct>     <fct> <dbl> <dbl> <chr>   <dbl> <dbl> <dbl> <dbl>
##  1 extreme   i1    0.4   0.516 conduct 0.6   0.712 0.356  51.8
##  2 extreme   i2    0.7   0.483 conduct 0.3   0.712 0.356  51.8
##  3 ambiguous i19   0.7   0.483 conduct 0.3   0.675 0.561  60.3
##  4 ambiguous i20   0.444 0.527 conduct 0.556 0.675 0.561  60.3
##  5 ambiguous i21   0.667 0.5   conduct 0.333 0.675 0.561  60.3
##  6 ambiguous i22   0.6   0.516 conduct 0.4   0.675 0.561  60.3
##  7 extreme   i12   0.222 0.441 conduct 0.778 0.712 0.356  51.8
##  8 extreme   i9    0.75  0.463 conduct 0.25  0.712 0.356  51.8
##  9 ambiguous i15   0.889 0.333 extract 0.111 0.889 0.533  67.8
## 10 ambiguous i16   0.6   0.516 extract 0.4   0.889 0.533  67.8
## # i 14 more rows
```

```
meansXtype
```

```
## # A tibble: 24 x 9
## # Groups:   pair, Xtype [6]
##    Xtype     item    acc    sd pair      err upper lower  mean
##    <fct>     <fct> <dbl> <dbl> <chr>   <dbl> <dbl> <dbl> <dbl>
##  1 ambiguous i13   0.556 0.527 refund  0.444 0.8   0.528  65
##  2 ambiguous i14   0.8   0.422 refund  0.2   0.8   0.528  65
##  3 ambiguous i15   0.889 0.333 extract 0.111 0.889 0.533  67.8
##  4 ambiguous i16   0.6   0.516 extract 0.4   0.889 0.533  67.8
##  5 extreme   i1    0.4   0.516 conduct 0.6   0.712 0.356  51.8
##  6 extreme   i2    0.7   0.483 conduct 0.3   0.712 0.356  51.8
##  7 extreme   i3    0.9   0.316 refund  0.1   0.788 0.528  66.2
##  8 extreme   i4    0.556 0.527 refund  0.444 0.788 0.528  66.2
##  9 ambiguous i17   0.333 0.5   extract 0.667 0.889 0.533  67.8
## 10 ambiguous i18   0.889 0.333 extract 0.111 0.889 0.533  67.8
## # i 14 more rows
```

Note: Plot below is Figure 4 in paper.

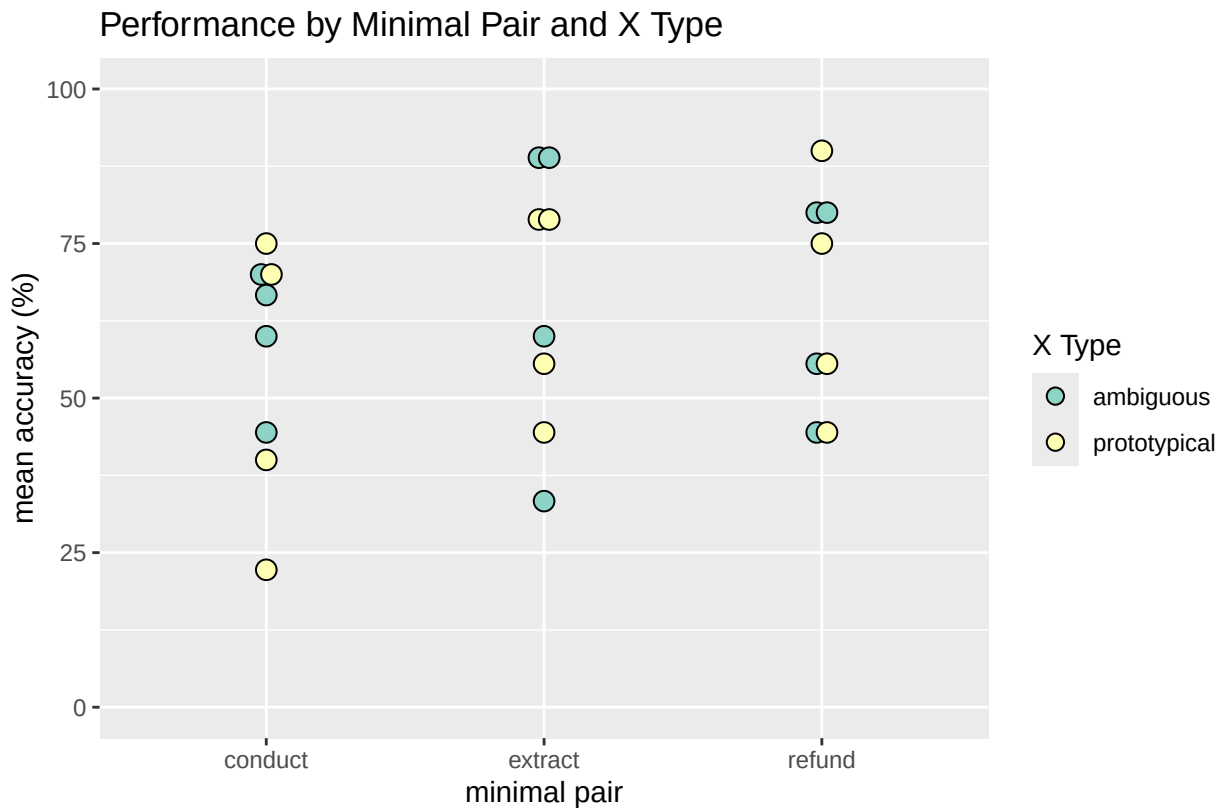
```
p <- ggplot(data = meansXtype, aes(x = pair, fill = Xtype, y = acc*100))+

    ## defining plot type and y-axis variable

  geom_dotplot(binaxis= "y", stackdir = "center", stackratio = .5, stackgroups = TRUE) +
```

```
  scale_fill_brewer(name = "X Type", labels = c("ambiguous", "prototypical"), palette="Set3") +

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

  ylim(0,100)
```

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

```
p
```

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



Note: Each dot represents participants' average performance on one item.

```
p <- ggplot(data = meansXtype2, aes(x = item, fill = Xtype, y = acc*100))+

    ## defining plot type and y-axis variable

  geom_dotplot(binaxis= "y", stackdir = "center", stackratio = .5, stackgroups = TRUE) +

  scale_fill_brewer(palette="Set3") +

  labs(y= "mean accuracy (%)", x = "minimal pair", title = "Accuracy by X type and Item") +

  ylim(0,100)
```

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

```
p
```

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