Modality effects in a signalling game: Accuracy

Contents

Intro

This script uses data compiled by *analyseData.R*.

Load libraries

```
library(lme4)
library(sjPlot)
library(ggplot2)
library(lattice)
library(influence.ME)
library(party)
library(dplyr)
```

Load data

```
d = read.csv("../../data/FinalSignalData.csv")
```

Work out number of turns in each trial.

```
# Number of turns in each trial
numTurns = tapply(d$turnString, d$trialString,
                  function(X){length(unique(X))})
d$numberOfTurns = numTurns[d$trialString]
```

Variable for length of first T1

```
T1L = tapply(d[d$turnType=="T1",]$turnLength,
             d[d$turnType=="T1",]$trialString, head, n=1)
d$T1Length = T1L[d$trialString]
```

```
d$T1Length[is.na(d$T1Length)] = mean(d$T1Length,na.rm=T)
d$T1Length.log = log(d$T1Length)
d$T1Length.log = d$T1Length.log - mean(d$T1Length.log)
```

Did matcher respond?

```
matcherResponds = tapply(d$turnType, d$trialString, function(X){
  any(X %in% c("T2","T4","T6","T8",'T10'))
})
d$matcherResponds = matcherResponds[d$trialString]
```

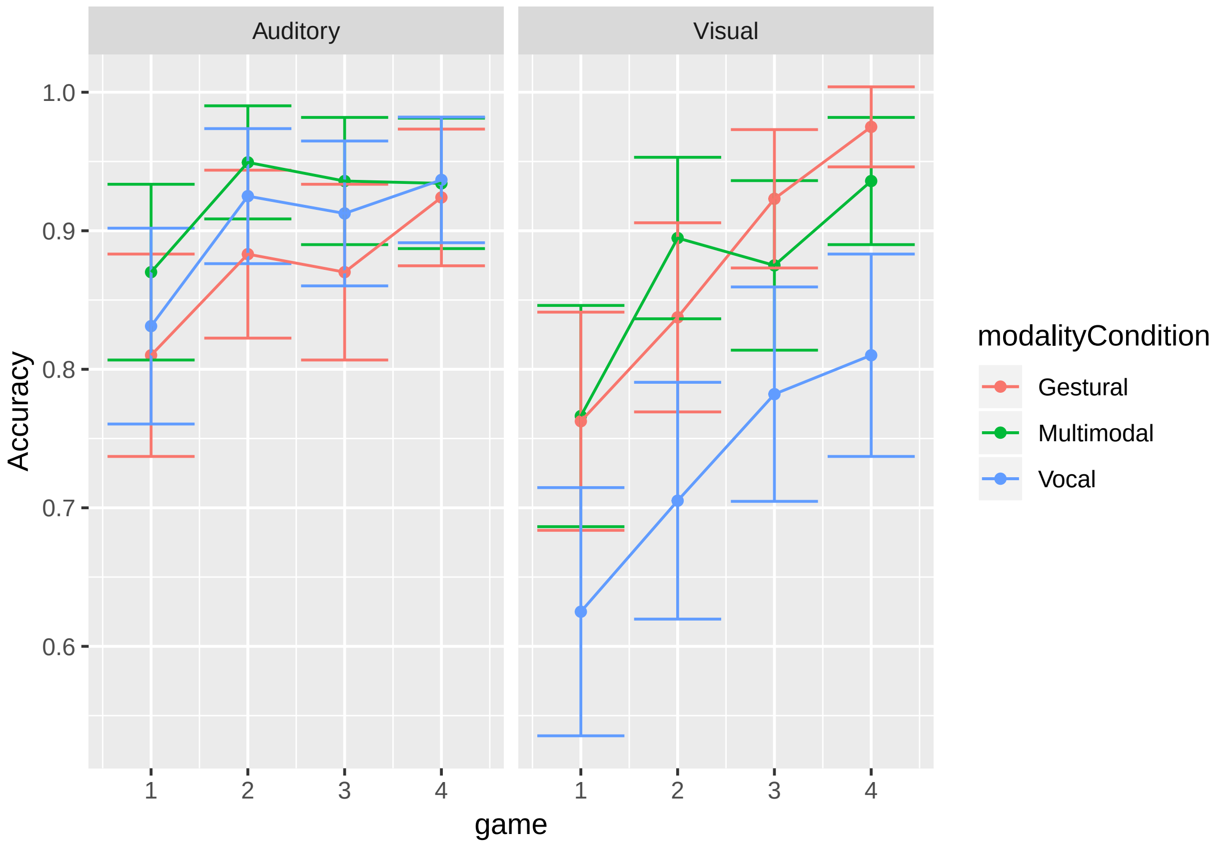
We don't need info on every signal in each turn, just the trial time. Keep only 1st signal in each trial.

```
d = d[!duplicated(d$trialString),]
```

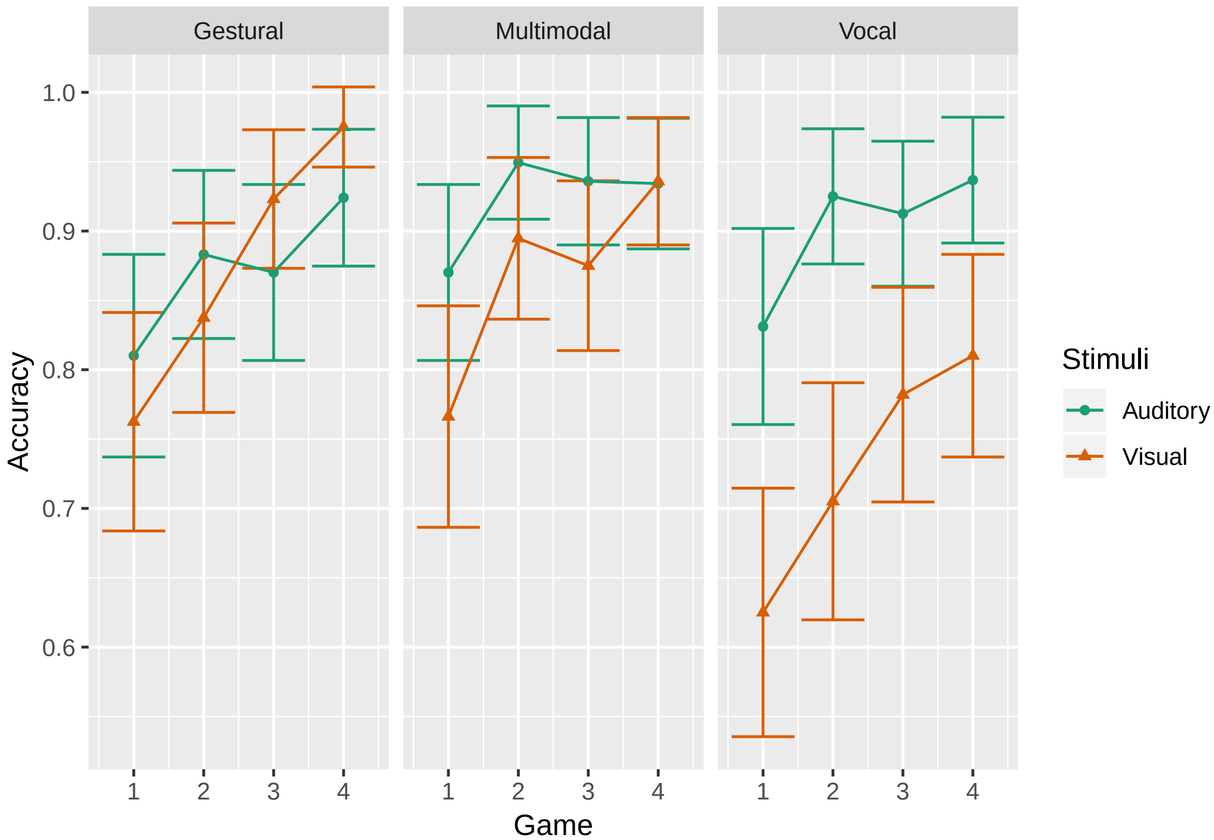
Descriptive stats

Here is a graph showing the distribution of accuracy by conditions:

```
summary = d %>%
  group_by(condition, modalityCondition,game) %>%
  summarise(Accuracy=mean(correct),
            sd=sd(correct),
            ci.w =          qnorm(0.95)*sd/sqrt(length(correct)),
            upper=Accuracy+ci.w,
            lower = Accuracy-ci.w)
summary$game = summary$game +1

summary$modalityCondition =
  factor(summary$modalityCondition,
         levels = c("visual",'multi','vocal'),
         labels=c("Gestural","Multimodal","Vocal"))

#ggplot(d, aes(x=trialTotal, y=as.numeric(correct),colour=modalityCondition)) +
#  geom_smooth() + facet_grid(.~condition)

#ggplot(d, aes(x=trialTotal, y=as.numeric(correct),colour=condition)) +
#  geom_smooth() + #facet_grid(.~modalityCondition)

ggplot(summary, aes(x=game, y=Accuracy, group=condition, colour=modalityCondition)) +
  geom_point() +
  geom_errorbar(aes(ymin=lower, ymax=upper)) +
  facet_grid(. ~ condition) +
  stat_summary(fun.y="mean", geom="line", aes(group=modalityCondition))
```



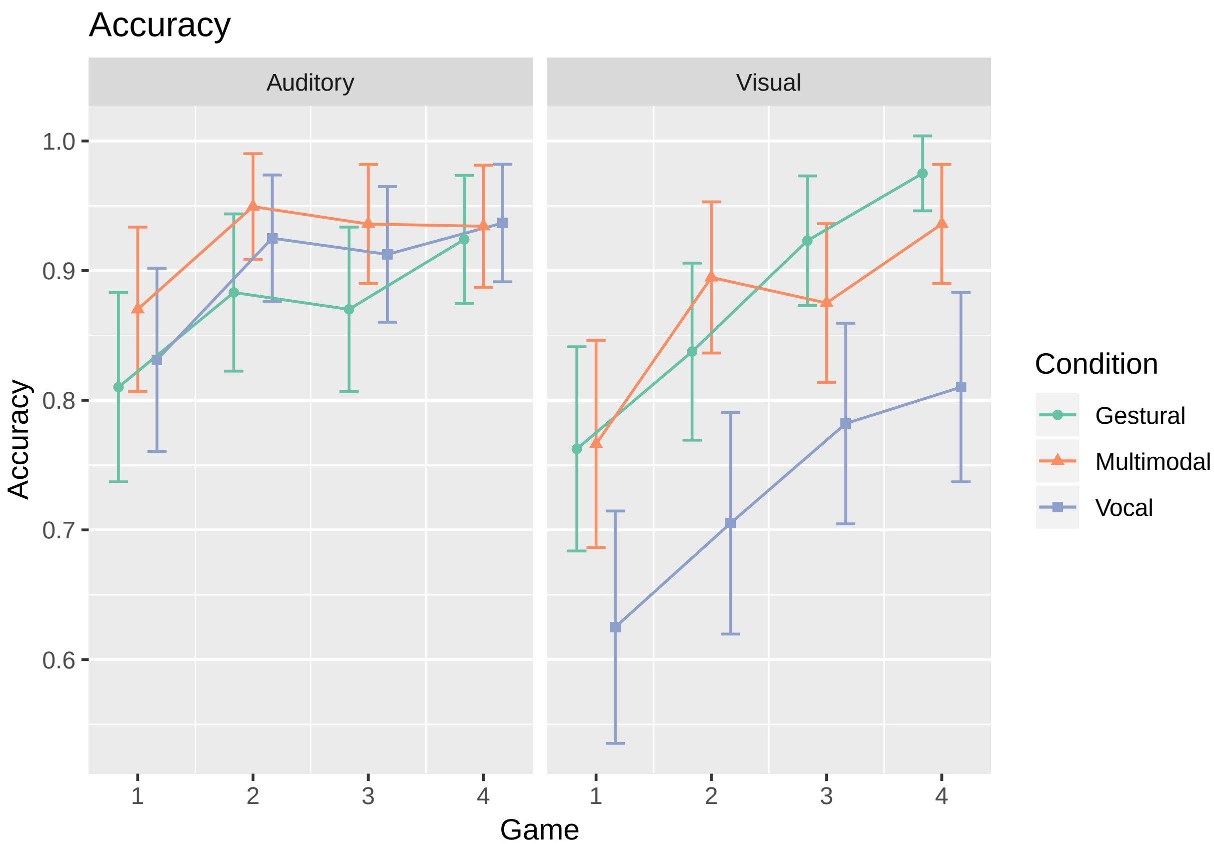

```
gx = ggplot(summary, aes(x=game, y=Accuracy, group=condition, colour=condition, shape=condition)) +
  geom_point() +
  geom_errorbar(aes(ymin=lower, ymax=upper)) +
  facet_grid(. ~ modalityCondition) +
  stat_summary(fun.y="mean", geom="line", aes(group=condition)) +
  scale_colour_brewer(palette="Dark2",name="Stimuli") +
  scale_shape_discrete(name="Stimuli") +
  xlab("Game")
gx
```




```r
pdf("../../results/graphs/Accuracy_gg.pdf",
    width = 5, height=3)
gx
dev.off()
```

```
## pdf
##   2
```

```r
pd = position_dodge(width=0.5)
gx1 = ggplot(summary, aes(x=game, y=Accuracy, group=condition, colour=modalityCondition)) +
  geom_errorbar(aes(ymin=lower, ymax=upper,group=modalityCondition), width=0.5,position = pd) +
  stat_summary(fun.y="mean", geom="line", aes(group=modalityCondition),position = pd) +
  geom_point(aes(group=modalityCondition,shape=modalityCondition),position=pd) +
  scale_colour_brewer(palette="Set2", name="Condition") +
  scale_shape(name="Condition") +
  ggtitle("Accuracy") +
  theme(panel.grid.major.x = element_blank()) +
  facet_grid(. ~ condition) +
  xlab("Game")

gx1
```




```
pdf("../../results/graphs/Accuracy_gg_alt.pdf",
    width = 5, height=3)
gx1
dev.off()
```

```
## pdf
##   2
```

Make a variable to represent proportion of games played:

```
# Make a variable that represents the number of trials played
d$trialTotal = d$trial + (d$game * (max(d$trial)+1))
# Convert to proportion of games played, so that estimates reflect change per game.
d$trialTotal = d$trialTotal / 16
# Center the trialTotal variable so intercept reflects after the first game
d$trialTotal = d$trialTotal
```

Average accuracy per dyad:

```
av.acc = tapply(d$correct, d$dyadNumber, function(X){sum(X)/length(X)})
mean(av.acc)
```

```
## [1] 0.8613356
```

Make a variable for which stimuli the players experienced first.

```
firstBlock = tapply(as.character(d$condition),d$dyadNumber,head,n=1)
d$firstBlock = as.factor(firstBlock[match(d$dyadNumber,names(firstBlock))])
```

Variable to indicate whether T1 is multimodal.

```
turnD = read.csv("../../data/Final_Turn_data.csv")
turnD = turnD[turnD$turnType=="T1",]
turnD = turnD[turnD$role == "Director",]
d$multimodal = turnD[match(d$trialString, turnD$trialString),]$turnModalityType == "multi"
d$multimodal[is.na(d$multimodal)] = F
```

Make a variable to represent proportion of games played:

```
# Make a variable that represents the number of trials played
d$trialTotal = d$trial + (d$game * (max(d$trial)+1))
# Convert to proportion of games played, so that estimates reflect change per game.
d$trialTotal = d$trialTotal / 16
# Center the trialTotal variable so intercept reflects after the first game
d$trialTotal = d$trialTotal - 2
```

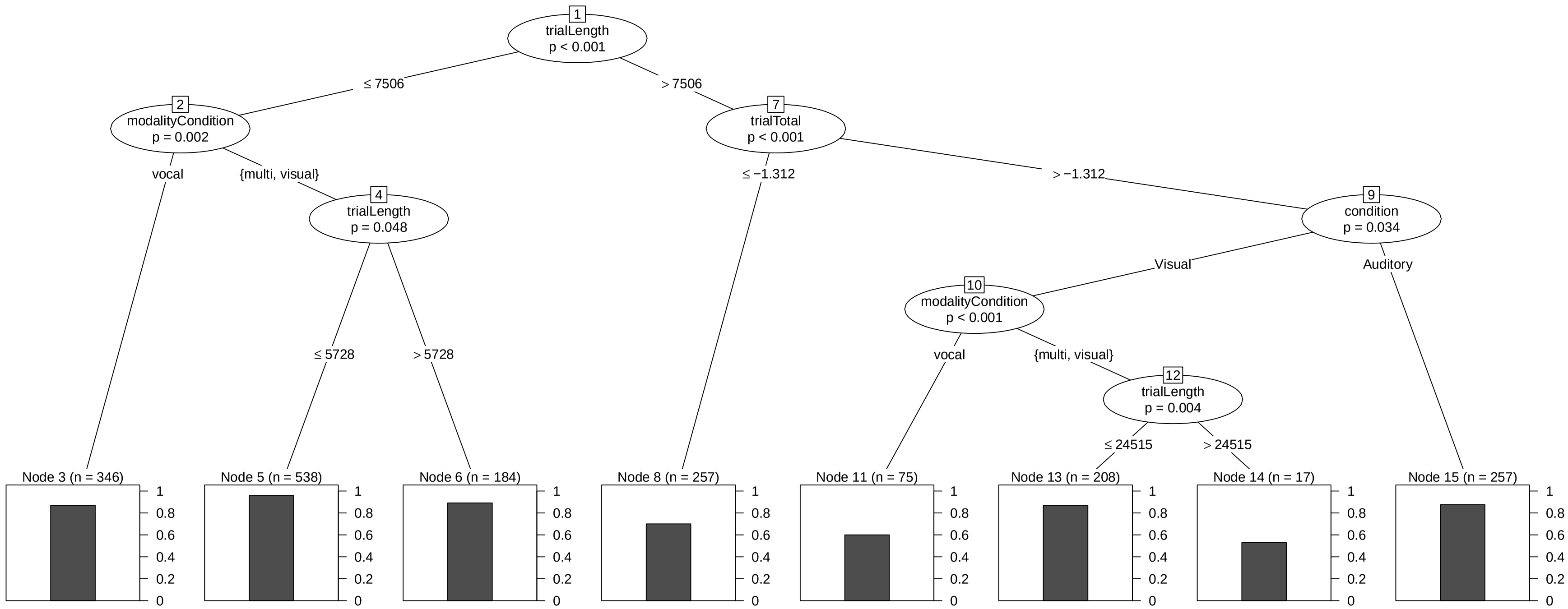
Transformed trial time.

```
d$trialLength.log = log(d$trialLength)
meanLogTrialLength = mean(d$trialLength.log)
d$trialLength.log = d$trialLength.log - meanLogTrialLength
```

Get an idea of the structure of the data from a binary tree:

```
cx = ctree(correct ~ modalityCondition + condition +
            trialTotal +
            trialLength +
            matcherResponds +
            matcherResponds +
            T1Length +
            multimodal+
            firstBlock,
          data=d)
```

```
plot(cx, terminal_panel=node_barplot(cx))
```



Mixed models

There are celing effects in the data, which reduces variance and makes model convergence difficult. Still, the following models converge relatively well.

```
# No fixed effects

gc = glmerControl(optimizer = "bobyqa" ,optCtrl = list(maxfun=50000))

m0 =  glmer(correct ~ 1 +
            (1 + condition |dyadNumber/playerId) +
            (1 + modalityCondition |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

```
game = glmer(correct ~ 1 +
            trialTotal +
            (1 + condition |dyadNumber/playerId) +
            (1 + modalityCondition |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

```
trialL = glmer(correct ~ 1 +
            trialTotal +
            trialLength.log+
            (1 + condition |dyadNumber/playerId) +
            (1 + modalityCondition |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

```
t1L = glmer(correct ~ 1 +
            trialTotal +
            trialLength.log +
            T1Length.log +
            (1 + condition |dyadNumber/playerId) +
            (1 + modalityCondition |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

```
multi = glmer(correct ~ 1 +
            trialTotal +
            trialLength.log +
            T1Length.log +
            multimodal+
            (1 + condition |dyadNumber/playerId) +
            (1 + modalityCondition |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

```r
mtchTrn = glmer(correct ~ 1 +
            trialTotal +
            trialLength.log +
            T1Length.log +
            multimodal+
            matcherResponds +
            (1 + condition |dyadNumber/playerId) +
            (1 + modalityCondition |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

```r
tMtchTr = glmer(correct ~ 1 +
            trialTotal +
            trialLength.log +
            T1Length.log +
            multimodal+
            matcherResponds +
            matcherResponds.cumulative +
            (1 + condition |dyadNumber/playerId) +
            (1 + modalityCondition |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

```r
con = glmer(correct ~ 1 + condition +
            trialTotal +
            trialLength.log +
            T1Length.log +
            multimodal+
            matcherResponds +
            matcherResponds.cumulative +
            (1 + condition |dyadNumber/playerId) +
            (1 + modalityCondition |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

```r
mod = glmer(correct ~ 1 + modalityCondition + condition +
            trialTotal +
            trialLength.log +
            T1Length.log +
            multimodal+
            matcherResponds +
            matcherResponds.cumulative +
            (1 + condition |dyadNumber/playerId) +
            (1 + modalityCondition |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

```
modXcon = glmer(correct ~ 1 + modalityCondition * condition +
            trialTotal +
            trialLength.log +
            T1Length.log +
            multimodal+
            matcherResponds +
            matcherResponds.cumulative +
            (1 + condition |dyadNumber/playerId) +
            (1 + modalityCondition |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

```
trialLXmo = glmer(correct ~ 1 + modalityCondition * condition +
            trialTotal +
            trialLength.log * modalityCondition+
            T1Length.log +
            multimodal+
            matcherResponds +
            matcherResponds.cumulative +
            (1 + condition |dyadNumber/playerId) +
            (1 + modalityCondition |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

```
t1LXmo = glmer(correct ~ 1 + modalityCondition * condition +
            trialTotal +
            trialLength.log * modalityCondition+
            T1Length.log *modalityCondition +
            multimodal+
            matcherResponds +
            matcherResponds.cumulative +
            (1 + condition |dyadNumber/playerId) +
            (1 + modalityCondition |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

```
tMaTXmo = glmer(correct ~ 1 + modalityCondition * condition +
            trialTotal +
            trialLength.log * modalityCondition+
            T1Length.log *modalityCondition +
            multimodal+
            matcherResponds +
            matcherResponds.cumulative +
              matcherResponds.cumulative:modalityCondition +
            (1 + condition |dyadNumber/playerId) +
            (1 + modalityCondition |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

```r
block = glmer(correct ~ 1 + modalityCondition * condition +
            trialTotal +
            trialLength.log * modalityCondition+
            T1Length.log *modalityCondition +
            multimodal+
            matcherResponds +
            matcherResponds.cumulative +
              matcherResponds.cumulative:modalityCondition +
            matcherResponds +
            firstBlock +
            (1 + condition |dyadNumber/playerId) +
            (1 + modalityCondition |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

Results

Compare the fit of the models:

```
modelComparison = anova(m0,con,mod,modXcon,
                        game, trialL,trialLXmo,
                        t1L, t1LXmo, mtchTrn, tMtchTr,tMaTXmo,
                        multi, block)
x = capture.output(modelComparison)
x[!grepl(": ",x)]
```

```
##  [1] "Models:"
##  [2] "          Df    AIC    BIC  logLik deviance   Chisq Chi Df Pr(>Chisq)    "
##  [3] "m0        13 1373.9 1445.9 -673.95   1347.9                              "
##  [4] "game      14 1322.0 1399.6 -647.02   1294.0 53.8678      1  2.144e-13 ***"
##  [5] "trialL    15 1285.0 1368.2 -627.53   1255.0 38.9791      1  4.284e-10 ***"
##  [6] "t1L       16 1286.5 1375.1 -627.23   1254.5  0.5884      1   0.443054    "
##  [7] "multi     17 1287.8 1382.0 -626.91   1253.8  0.6440      1   0.422265    "
##  [8] "mtchTrn   18 1288.9 1388.6 -626.46   1252.9  0.9003      1   0.342703    "
##  [9] "tMtchTr   19 1283.8 1389.1 -622.90   1245.8  7.1172      1   0.007635 ** "
## [10] "con       20 1283.3 1394.2 -621.68   1243.3  2.4515      1   0.117413    "
## [11] "mod       22 1286.1 1408.0 -621.06   1242.1  1.2277      2   0.541255    "
## [12] "modXcon   24 1286.1 1419.0 -619.03   1238.1  4.0598      2   0.131346    "
## [13] "trialLXmo 26 1289.1 1433.1 -618.55   1237.1  0.9690      2   0.615996    "
## [14] "t1LXmo    28 1292.2 1447.3 -618.08   1236.2  0.9396      2   0.625143    "
## [15] "tMaTXmo   30 1294.1 1460.3 -617.07   1234.1  2.0168      2   0.364794    "
## [16] "block     31 1295.8 1467.5 -616.90   1233.8  0.3400      1   0.559806    "
## [17] "---"
```

Final model with only significant variables:

```
finalModel = glmer(correct ~ 1 +
            modalityCondition * condition +
            trialTotal +
            trialLength.log +
            matcherResponds.cumulative +
            (1 + condition | dyadNumber/playerId) +
            (1 + modalityCondition |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

Model estimates:

```
summary(finalModel)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## correct ~ 1 + modalityCondition * condition + trialTotal + trialLength.log +
##     matcherResponds.cumulative + (1 + condition | dyadNumber/playerId) +
##     (1 + modalityCondition | itemId)
##    Data: d
## Control: gc
##
```

```
##      AIC      BIC   logLik deviance df.resid
##   1281.3   1397.7   -619.7   1239.3     1861
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -9.2650  0.1006  0.2159  0.3840  2.1256
##
## Random effects:
##  Groups              Name                     Variance Std.Dev. Corr
##  playerId:dyadNumber (Intercept)              0.6991   0.8361
##                      conditionVisual          0.3243   0.5695   -0.92
##  itemId              (Intercept)              1.6445   1.2824
##                      modalityConditionvisual  0.7057   0.8401   -0.52
##                      modalityConditionvocal   0.5105   0.7145   -0.54 -0.44
##  dyadNumber          (Intercept)              0.1531   0.3913
##                      conditionVisual          0.3331   0.5771   -0.87
## Number of obs: 1882, groups:
## playerId:dyadNumber, 30; itemId, 16; dyadNumber, 15
##
## Fixed effects:
##                                         Estimate Std. Error z value
## (Intercept)                              3.07427    0.65510   4.693
## modalityConditionvisual                 -0.60237    0.67303  -0.895
## modalityConditionvocal                   0.09082    0.66262   0.137
## conditionVisual                         -0.94005    0.80945  -1.161
## trialTotal                               0.21387    0.07816   2.736
## trialLength.log                         -1.02009    0.15873  -6.427
## matcherResponds.cumulative               0.11509    0.04813   2.391
## modalityConditionvisual:conditionVisual  0.90728    0.76320   1.189
## modalityConditionvocal:conditionVisual  -0.97050    0.73095  -1.328
##                                         Pr(>|z|)
## (Intercept)                             2.69e-06 ***
## modalityConditionvisual                  0.37078
## modalityConditionvocal                   0.89098
## conditionVisual                          0.24550
## trialTotal                               0.00621 **
## trialLength.log                         1.31e-10 ***
## matcherResponds.cumulative               0.01679 *
## modalityConditionvisual:conditionVisual  0.23453
## modalityConditionvocal:conditionVisual   0.18427
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##               (Intr) mdltyCndtnvs mdltyCndtnvc cndtnV trlTtl trlLn.
## mdltyCndtnvs  -0.608
## mdltyCndtnvc  -0.622  0.357
## conditinVsl   -0.716  0.452        0.417
## trialTotal     0.115  0.006       -0.081        0.018
## trlLngth.lg    0.021 -0.050       -0.074       -0.049  0.379
## mtchrRspnd.   -0.132 -0.083        0.108       -0.107 -0.402 -0.096
## mdltyCndtnvs:V  0.461 -0.760       -0.236       -0.587 -0.031  0.019
## mdltyCndtnvc:V  0.461 -0.276       -0.763       -0.593 -0.020  0.109
##               mtchR. mdltyCndtnvs:V
```

```
## mdltyCndtnvs
## mdltyCndtnvc
## conditinVsl
## trialTotal
## trlLngth.lg
## mtchrRspnd.
## mdltyCndtnvs:V  0.118
## mdltyCndtnvc:V  0.128  0.259
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```
# number of correctly categorised trials
sum((predict(finalModel)>0) == d$correct)/nrow(d)
```

```
## [1] 0.8687566
```

Surprisingly, the interaction between modality and stimulus condition is not significant. In comparison, in a model without random slopes, the interaction is significant:

```
finalModel.simple = glmer(correct ~ 1 +
            modalityCondition * condition +
            trialTotal +
            trialLength.log +
            matcherResponds.cumulative +
            (1  |dyadNumber/playerId) +
            (1  |itemId) ,
          data=d, family=binomial,
          control = gc)
summary(finalModel.simple)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## correct ~ 1 + modalityCondition * condition + trialTotal + trialLength.log +
##     matcherResponds.cumulative + (1 | dyadNumber/playerId) +
##     (1 | itemId)
##    Data: d
## Control: gc
##
##      AIC      BIC   logLik deviance df.resid
##   1296.4   1362.9   -636.2   1272.4     1870
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -9.6994  0.1243  0.2386  0.4062  1.9291
##
## Random effects:
##  Groups              Name        Variance Std.Dev.
##  playerId:dyadNumber (Intercept) 0.22507  0.4744
##  itemId              (Intercept) 0.85424  0.9242
##  dyadNumber          (Intercept) 0.04445  0.2108
## Number of obs: 1882, groups:
## playerId:dyadNumber, 30; itemId, 16; dyadNumber, 15
##
## Fixed effects:
```

```
##                                         Estimate Std. Error z value
## (Intercept)                              2.66579    0.46029   5.792
## modalityConditionvisual                 -0.36893    0.39095  -0.944
## modalityConditionvocal                   0.26560    0.40631   0.654
## conditionVisual                         -0.55651    0.56042  -0.993
## trialTotal                               0.21171    0.07600   2.786
## trialLength.log                         -1.02939    0.15120  -6.808
## matcherResponds.cumulative               0.08419    0.04374   1.925
## modalityConditionvisual:conditionVisual  0.66376    0.39552   1.678
## modalityConditionvocal:conditionVisual  -1.16638    0.39524  -2.951
##                                         Pr(>|z|)
## (Intercept)                             6.98e-09 ***
## modalityConditionvisual                  0.34534
## modalityConditionvocal                   0.51332
## conditionVisual                          0.32070
## trialTotal                               0.00534 **
## trialLength.log                         9.88e-12 ***
## matcherResponds.cumulative               0.05424 .
## modalityConditionvisual:conditionVisual  0.09331 .
## modalityConditionvocal:conditionVisual   0.00317 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##               (Intr) mdltyCndtnvs mdltyCndtnvc cndtnV trlTtl trlLn.
## mdltyCndtnvs   -0.445
## mdltyCndtnvc   -0.493  0.510
## conditinVsl    -0.622  0.258        0.208
## trialTotal      0.160 -0.008       -0.141        0.011
## trlLngth.lg     0.037 -0.102       -0.141       -0.066  0.387
## mtchrRspnd.    -0.210 -0.119        0.214       -0.113 -0.386 -0.121
## mdltyCndtnvs:V  0.244 -0.574       -0.278       -0.409 -0.046  0.041
## mdltyCndtnvc:V  0.242 -0.374       -0.556       -0.417 -0.006  0.212
##               mtchR. mdltyCndtnvs:V
## mdltyCndtnvs
## mdltyCndtnvc
## conditinVsl
## trialTotal
## trlLngth.lg
## mtchrRspnd.
## mdltyCndtnvs:V  0.204
## mdltyCndtnvc:V  0.171  0.583
```

By model comparison, we should prefer the model with random slopes:

```
anova(finalModel.simple,finalModel)
```

```
## Data: d
## Models:
## finalModel.simple: correct ~ 1 + modalityCondition * condition + trialTotal + trialLength.log +
## finalModel.simple:     matcherResponds.cumulative + (1 | dyadNumber/playerId) +
## finalModel.simple:     (1 | itemId)
## finalModel: correct ~ 1 + modalityCondition * condition + trialTotal + trialLength.log +
## finalModel:     matcherResponds.cumulative + (1 + condition | dyadNumber/playerId) +
## finalModel:     (1 + modalityCondition | itemId)
```

```
##                   Df    AIC    BIC  logLik deviance  Chisq Chi Df
## finalModel.simple 12 1296.4 1362.9 -636.22   1272.4
## finalModel        21 1281.3 1397.7 -619.67   1239.3 33.086      9
##                   Pr(>Chisq)
## finalModel.simple
## finalModel          0.000129 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This suggests that, while accuracy is lower for visual stimuli in the vocal condition, the difference is not greater than might be expected by random (slope) variation between dyads and items.

We also show that a model with more random slopes is essentially identical to the simpler random slopes model:

```
finalModel.full = glmer(correct ~ 1 +
            modalityCondition * condition +
            trialTotal +
            trialLength.log +
            matcherResponds.cumulative +
            (1 + condition + trialTotal + trialLength.log +
               matcherResponds.cumulative |dyadNumber/playerId) +
            (1 + modalityCondition + trialTotal + trialLength.log +
               matcherResponds.cumulative |itemId) ,
          data=d, family=binomial,
          control = gc)
```

```
## boundary (singular) fit: see ?isSingular
```

```
cor(fixef(finalModel.full),fixef(finalModel))
```

```
## [1] 0.9939008
```

Plot the fixed effects

Relabel the effects:

```
feLabels = matrix(c(
"(Intercept)"             ,"Intercept"      , NA,
"modalityConditionvisual" ,"Visual modality", "mod",
"modalityConditionvocal"  , "Acoustic modality", "mod",
"conditionVisual" , "Visual stimuli","con",
"trialTotal"             , "Game","game",
"modalityConditionvisual:conditionVisual"  , "Visual modality:Visual stimuli", "modXcon",
"modalityConditionvocal:conditionVisual" , "Acoustic modality:Visual stimuli","modXcon",
"firstBlockVisual","Visual stims first","block",
"trialLength.log","Trial length","trialL",
"modalityConditionvisual:trialLength.log","Visual modality:Trial length",'trialLXmo',
"modalityConditionvocal:trialLength.log","Acoustic modality:Trial length",'trialLXmo',
"multimodalTRUE","Multimodal T1","multi",
"trialLength.log",'Trial Length','trialL',
"T1Length.log","T1 length","t1L",
"modalityConditionvisual:T1Length.log","T1 length:Visual modality","t1LXmo",
"modalityConditionvocal:T1Length.log","T1 length:Acoustic modality","t1LXmo",
"matcherRespondsTRUE","Matcher Responds",'mtchTrn',
"matcherResponds.cumulative","Total interaction","tMtchTr",
```

```r
"modalityConditionvisual:matcherResponds.cumulative","Total interaction:Visual Modality","tMaTXmo",
"modalityConditionvocal:matcherResponds.cumulative","Total interaction:Vocal Modality","tMaTXmo"
), ncol=3, byrow = T)
feLabels1 = as.vector(feLabels[match(names(fixef(finalModel)),feLabels[,1]),1])
feLabels2 = as.vector(feLabels[match(names(fixef(finalModel)),feLabels[,1]),2])
feModel = as.vector(feLabels[match(names(fixef(finalModel)),feLabels[,1]),3])

sig = modelComparison$`Pr(>Chisq)`
names(sig) = rownames(modelComparison)

sig.data = data.frame(
  estimate = fixef(finalModel),
  y=1:length(fixef(finalModel)),
  sig=sig[feModel])
sig.data$fade = sig.data$sig > 0.05
```

Plot the strength of the fixed effects:

```r
plot_model(finalModel, 'est',
         show.intercept = T,
         sort.est=NULL,
         axis.labels = feLabels2[2:length(feLabels2)],
         axis.title="Odds of correct selection",
         colors = c(1,1),
         show.values = F,
         show.p = T,
         string.interc="Intercept",
         prnt.plot = F)
```

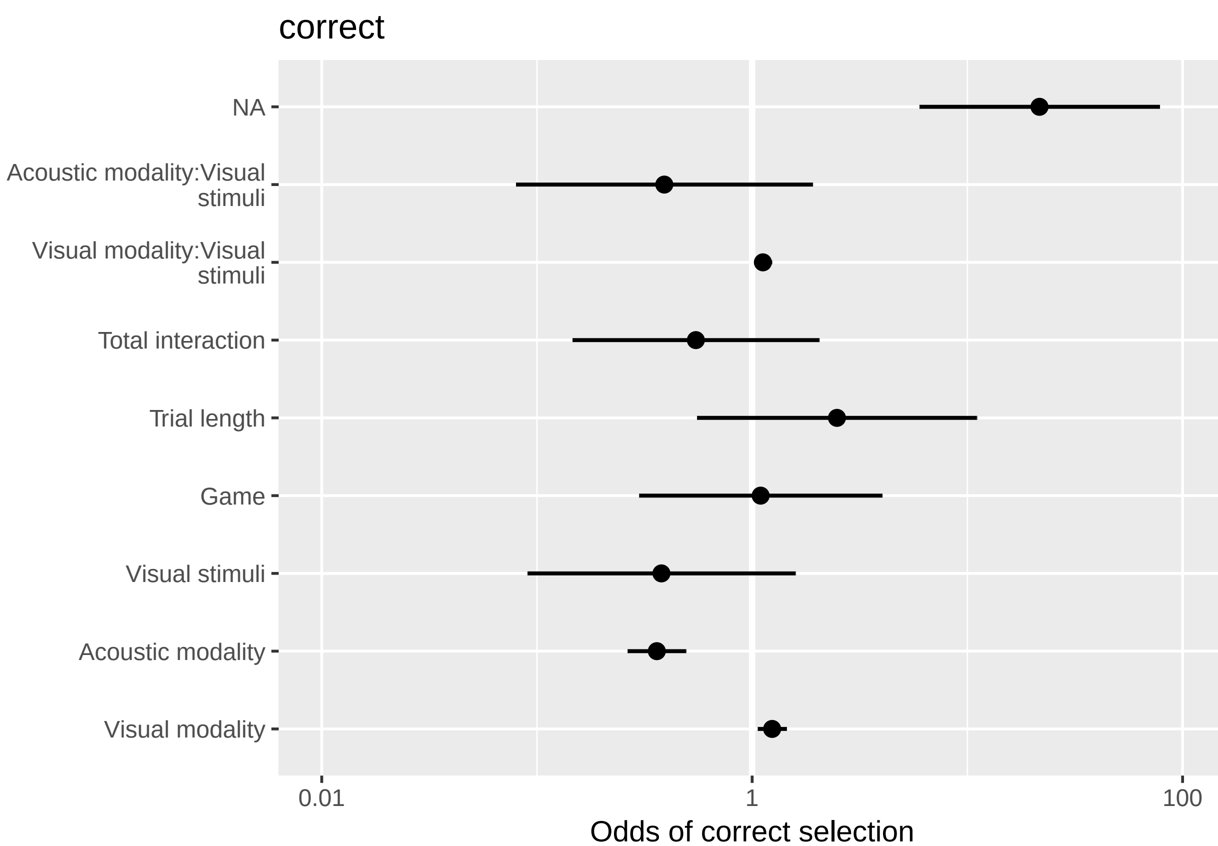


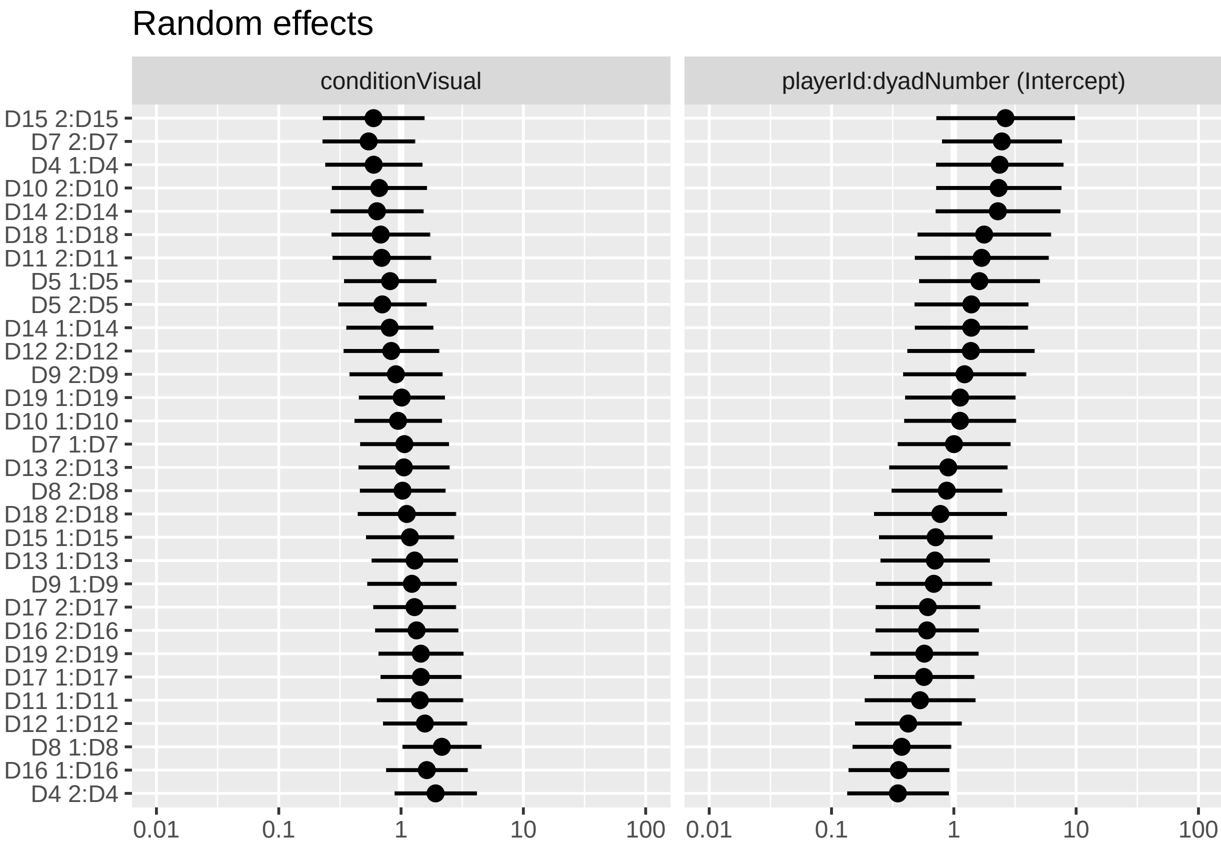
Table of results

```
x = as.data.frame(summary(finalModel)$coef)
mc = as.data.frame(modelComparison)
finalRes= cbind(x,mc[feModel,])
write.csv(finalRes, "../../results/tables/Accuracy_FixedEffects.csv")
```

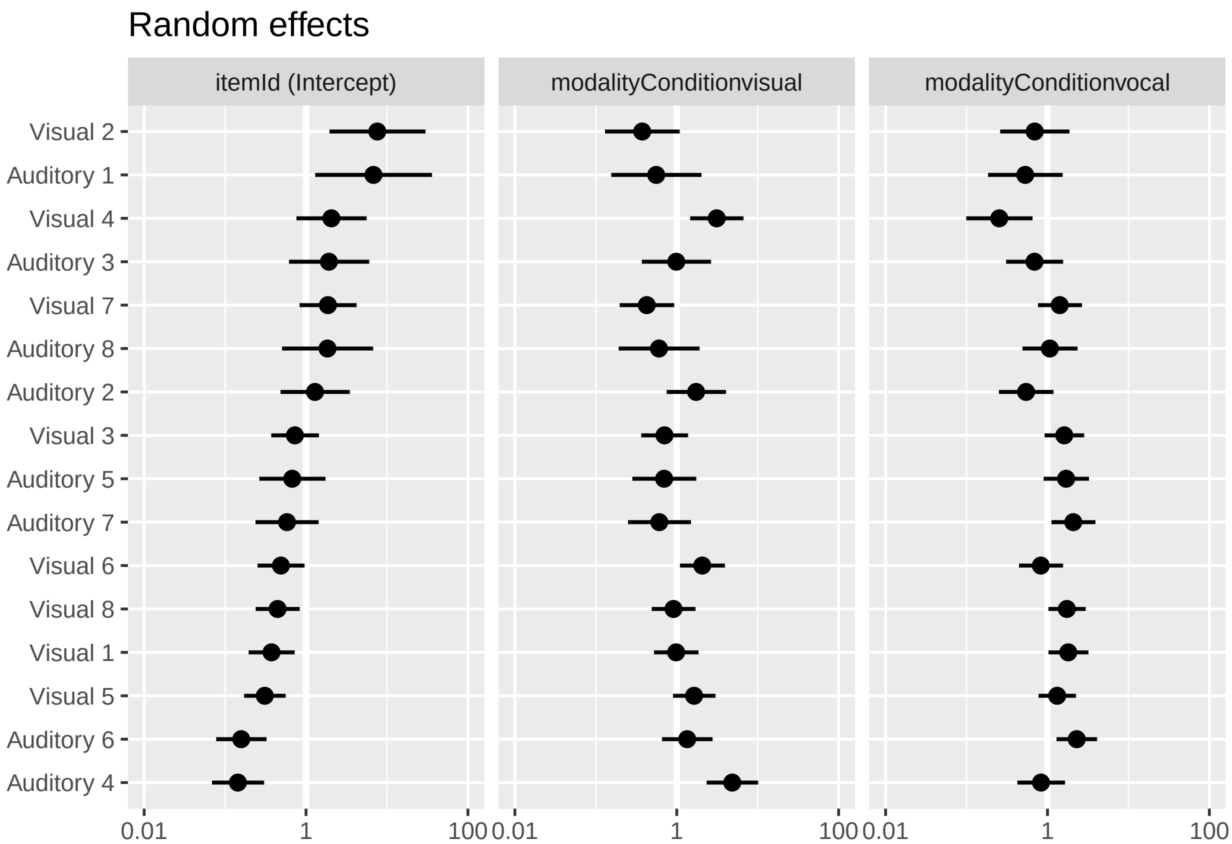
Random effects

```r
plot_model(finalModel,'re', sort.est ="(Intercept)",
          colors= c(1,1))
```

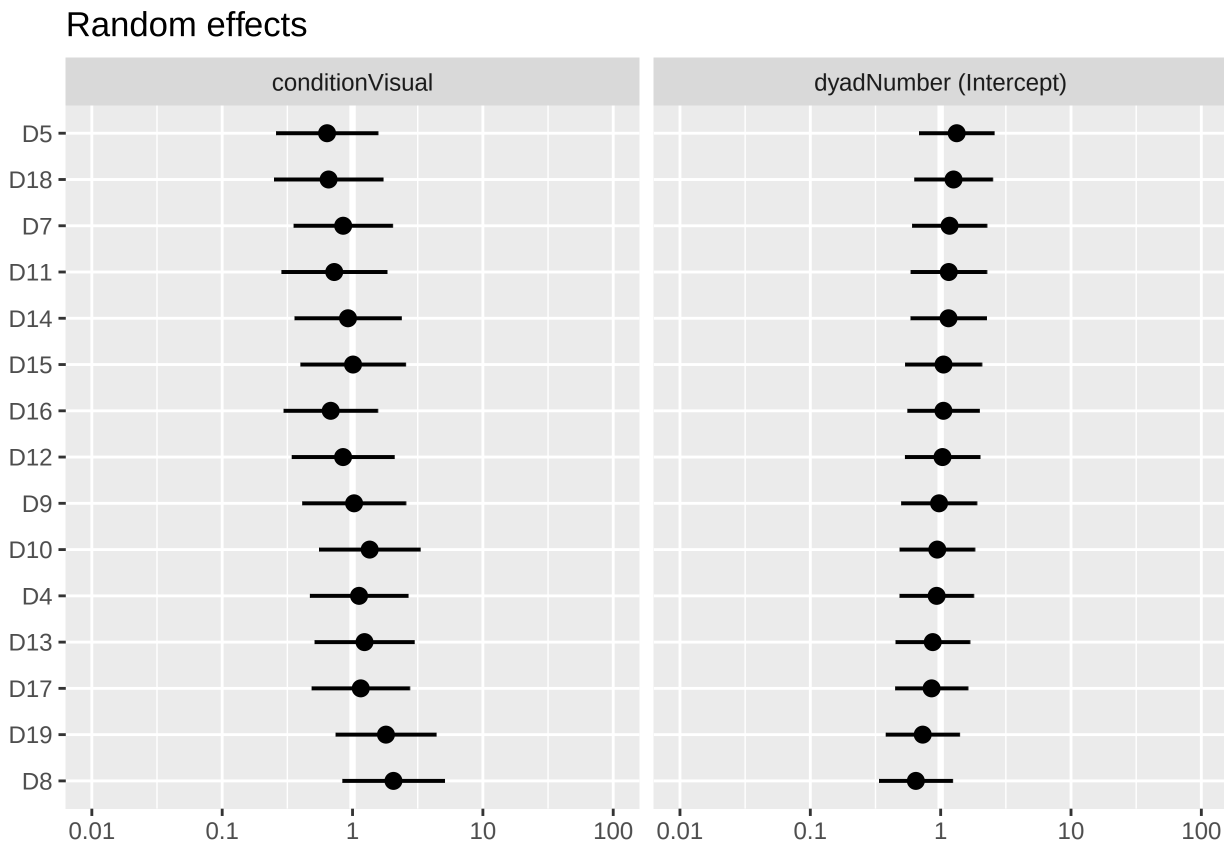
```
## [[1]]
```




```
##
## [[2]]
```


Random effects
itemId (Intercept)
modalityConditionvisual
modalityConditionvocal
Visual 2
Auditory 1
Visual 4
Auditory 3
Visual 7
Auditory 8
Auditory 2
Visual 3
Auditory 5
Auditory 7
Visual 6
Visual 8
Visual 1
Visual 5
Auditory 6
Auditory 4
0.01
1
100

```
##
## [[3]]
```


Random effects
conditionVisual
dyadNumber (Intercept)
D5
D18
D7
D11
D14
D15
D16
D12
D9
D10
D4
D13
D17
D19
D8
0.01
0.1
1
10
100

Plots

```
gx = get_model_data(finalModel, 'eff', vars=c("matcherResponds.cumulative"),
               title = '', prnt.plot = F,
               show.ci=T, facet.grid = F)
gxx = ggplot(gx$matcherResponds.cumulative,aes(x=x,y=predicted)) +
  geom_line(size=1.5) +
  geom_ribbon(aes(ymin=conf.low,ymax=conf.high),alpha=0.3) +
  xlab("Number of previous trials where\nmatcher responded") +
  ylab("Probability of correct choice") +
  #scale_x_continuous(breaks = c(0,5,10,15)) +
  coord_cartesian(xlim=c(0,15))
pdf("../../results/graphs/CumulativeMatcherTurns.pdf",
    width=4,height=4)
gxx
dev.off()
gxx
```

Similar plot, showing raw data and how number of trials and cumulative number of matcher responses relate for correct and incorrect guesses. It shows that correct guesses tend to be preceded by more matcher responses, especially late in the experiment.

In the plot below, we plot the model predictions (line and ribbon) against the real probability of being correct (points with error bars representing 95% confidence intervals according to the binomial test). The effect size for the model predictions is less extreme, since some of the variance is captured by number of trials.

```
cuts = c(0,1,5,9,13,17)
d$matcherResponds.cumulative.cat = cut(d$matcherResponds.cumulative,cuts,include.lowest = T)
midpoints = c(0,cuts[2:length(cuts)]+(diff(cuts[2:length(cuts)])[1])/2)
cumClust = data.frame()
for(i in 1:length(levels(d$matcherResponds.cumulative.cat))){
  mp = midpoints[i]
  cat = levels(d$matcherResponds.cumulative.cat)[i]
  tx = c(sum(d[d$matcherResponds.cumulative.cat==cat,]$correct),
         sum(!d[d$matcherResponds.cumulative.cat==cat,]$correct))
  bt = binom.test(tx)
  cumClust = rbind(cumClust,
          c(mp,bt$estimate,bt$conf.int))
}
names(cumClust) = c("x",'predicted','low','high')
pdf("../../results/graphs/CumulativeMatcherTurns_withRawData.pdf")
ggplot(gx$matcherResponds.cumulative,aes(x=x,y=predicted)) +
  geom_line(size=1.5) +
  geom_ribbon(aes(ymin=conf.low,ymax=conf.high),alpha=0.3) +
  geom_point(data=cumClust,mapping=aes(x=x,y=predicted),size=3) +
  geom_errorbar(data=cumClust,mapping=aes(x=x,ymin=low,ymax=high)) +
  xlab("Number of previous trials where\nmatcher responded") +
  ylab("Probability of correct choice")
dev.off()
```

Variance explained

Total variance explained by the model: Calculated by pseudo R squared method from the *MuMIn* package to calculate the variance explained by fixed effects and random effects in a model (Nakagawa & Schielzeth 2013).

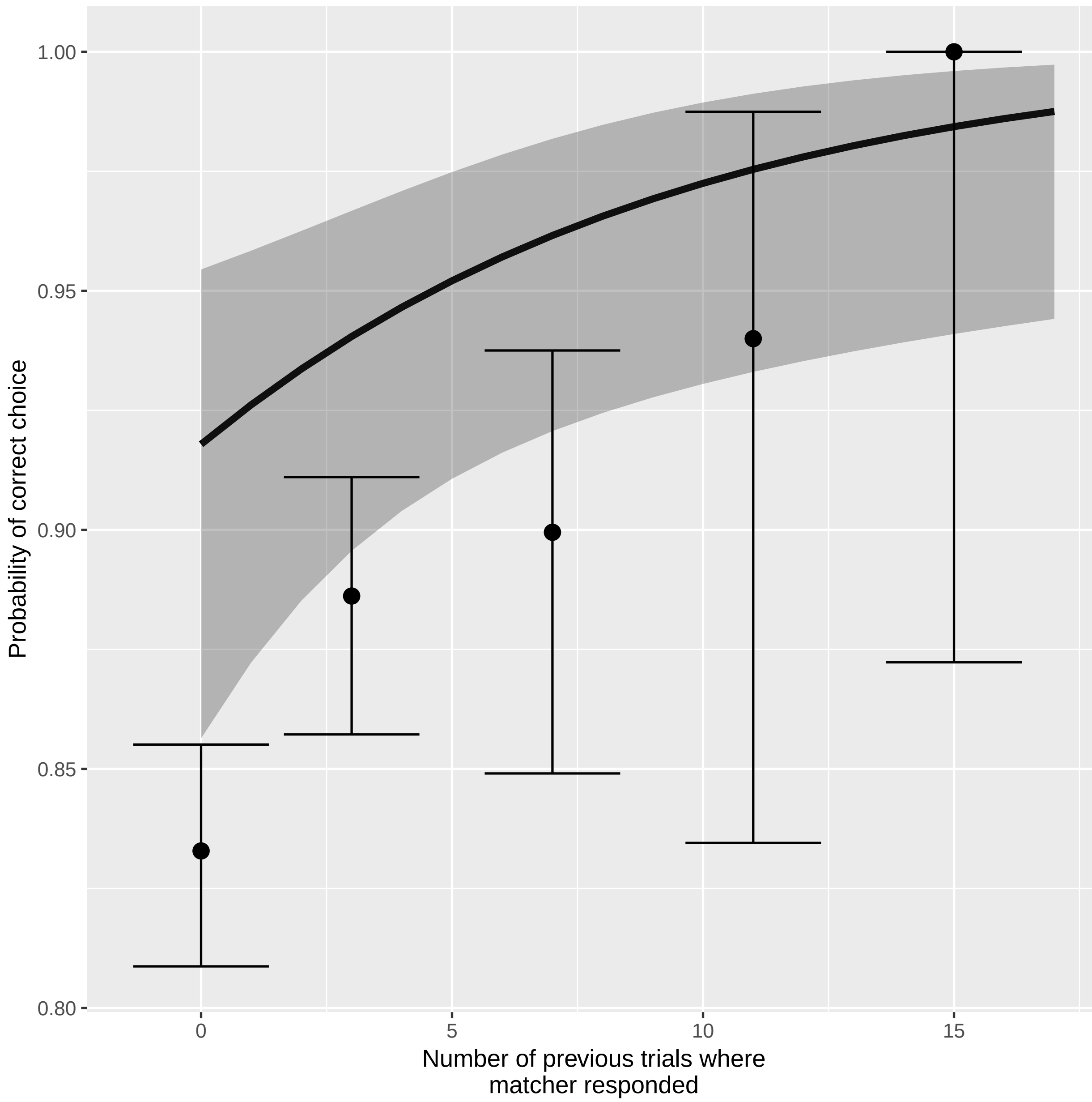

1.00
0.95
0.90
0.85
0.80
Probability of correct choice
0
5
10
15
Number of previous trials where
matcher responded

Figure 1:

```r
r.squaredGLMM.binom = function(object){
fam <- family(object)
  fitted <- (model.matrix(object) %*% MuMIn:::.numfixef(object))[,1L]
  varFE <- var(fitted)
  mmRE <- MuMIn:::.remodmat(object)
  vc <- MuMIn:::.varcorr(object)
  varRE <- MuMIn:::.varRESum(vc, mmRE)
  familyName <- fam$family
  null <- MuMIn:::.nullFitRE(object)
  fixefnull <- unname(MuMIn:::.numfixef(null))
  vt <- MuMIn:::.varRESum(MuMIn:::.varcorr(null), mmRE)
  pmean <- fam$linkinv(fixefnull - 0.5 * vt * tanh(fixefnull *
                                    (1 + 2 * exp(-0.5 * vt))/6))
  MuMIn:::r2glmm(fam, varFE, varRE, pmean = pmean)
}
```

```r
r.squaredGLMM.binom(finalModel)
```

```
##                   R2m       R2c
## theoretical 0.1818443 0.4805211
## delta       0.1022047 0.2700745
```

```r
fee = r.squaredGLMM.binom(finalModel)[1,1]
tee = r.squaredGLMM.binom(finalModel)[1,2]
ree = tee-fee
```

Fixed effects explain 18.18% of the variance. Total varaince explained = 48.05%. (random effects = 29.87).

For each model in the bottom-up procedure, we then calculate the increase in variance explained. This is an estiamte of how much variance a particular variable accounts for.

```r
mList = list("m0"=m0,"con"=con,"mod"=mod,"modXcon"=modXcon,
  "game"=game,"trialL"=trialL,"trialLXmo"=trialLXmo,
  "t1L"=t1L,"t1LXmo"=t1LXmo,"mtchTrn"=mtchTrn,"tMtchTr"=tMtchTr,
  "tMaTXmo"=tMaTXmo,"multi"=multi,"block"=block)
mList = mList[rownames(modelComparison)]
varExplained = sapply(mList,r.squaredGLMM.binom)
varExplained.fixed = diff(varExplained[1,])

t(t(varExplained.fixed[c("mod","con","modXcon",
                      "game","trialL","mtchTrn")]))
```

```
##                  [,1]
## mod      0.0136149651
## con      0.0436682580
## modXcon -0.0001404588
## game     0.0510865390
## trialL   0.0504628919
## mtchTrn  0.0035762305
```

Summary results

```r
signif(finalRes[,c("Estimate","Std. Error","z value",
                   "Pr(>|z|)",'Chisq',"Pr(>Chisq)")],2)
```

```
##                                         Estimate Std. Error z value
## (Intercept)                                3.100      0.660    4.70
## modalityConditionvisual                   -0.600      0.670   -0.90
## modalityConditionvocal                     0.091      0.660    0.14
## conditionVisual                           -0.940      0.810   -1.20
## trialTotal                                 0.210      0.078    2.70
## trialLength.log                           -1.000      0.160   -6.40
## matcherResponds.cumulative                 0.120      0.048    2.40
## modalityConditionvisual:conditionVisual    0.910      0.760    1.20
## modalityConditionvocal:conditionVisual    -0.970      0.730   -1.30
##                                         Pr(>|z|) Chisq Pr(>Chisq)
## (Intercept)                              2.7e-06    NA         NA
## modalityConditionvisual                  3.7e-01   1.2    5.4e-01
## modalityConditionvocal                   8.9e-01   1.2    5.4e-01
## conditionVisual                          2.5e-01   2.5    1.2e-01
## trialTotal                               6.2e-03  54.0    2.1e-13
## trialLength.log                          1.3e-10  39.0    4.3e-10
## matcherResponds.cumulative               1.7e-02   7.1    7.6e-03
## modalityConditionvisual:conditionVisual  2.3e-01   4.1    1.3e-01
## modalityConditionvocal:conditionVisual   1.8e-01   4.1    1.3e-01
```