

Modality effects in a signalling game: Efficiency

Contents

Intro	1
Load libraries	2
Load data	2
Descriptive stats	2
Mixed models	9
Results	15
Plot the fixed effects	20
Random effects	25
Relevel factors to see other comparisons	27
Variance explained	33
Alternative models	35
Model for only accurate trials	35
Model for only multimodal trials	37
Minimal model	40
Alternative distribution	41
Summary results	46

Intro

The main data used in this analysis comes from `../data/FinalSignalData.csv` (compiled by `analyseData.R`). Each row represents one signal, but this script only keeps one signal per trial, and the rest of the analysis is on the trial-by-trial level. The variables in the data are as follows (some are calculated in the script below):

- X: ID
- filename: Filename of the ELAN file
- dyadNumber: ID of the participant dyad
- condition: Stimuli type (Auditory or Visual)
- game: Game number (0-3)
- trial: Trial number (0-15)
- target: Target stimuli shown to the director
- choice: Meaning chosen by the matcher
- correct: True if the matcher's choice is correct
- trialStart, trialEnd, trialLength: Start, end and length of trial in milliseconds
- trialValue: A unique string that represents data from the trial. Numbers in the curly brackets represent the choices given to the matcher
- startOfNextTrial: Timestamp for next trial, used in processing the data.
- turnStart, turnEnd, turnLength: the start, end and length of the turn in milliseconds.
- signalStart, signalEnd, signalLength: the start, end and length of the signal.
- signalType: Annotation value in ELAN, not meaningful
- trialString: Unique string to identify trial
- modalityCondition: The condition for the dyad (multi= multimodal, visual=gesture only, vocal=vocal only)

- playerId: Unique ID for the participant producing the signal
- itemId: Unique ID for the target stimulus
- turnString: Unique ID for the turn
- matcherResponds: Does matcher take a turn in this trial?
- matcherResponds.cumulative: The (scaled) number of previous trials that a has responded.
- T1Length, T1Length.log: Length and log length of the director's first turn.
- trialTotal: Number of trials played so far, scaled (and centered) to represent number of games played.
- firstBlock: Block order
- incorrect: Was the matcher's choice incorrect?
- multimodal: Was the director's first turn multimodal?

Load libraries

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

The sjPlot library was updated during this investigation, removing various functions. They are reinstated here:

```
sjp.lmer = plot_model
```

Load data

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

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)
```

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

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
```

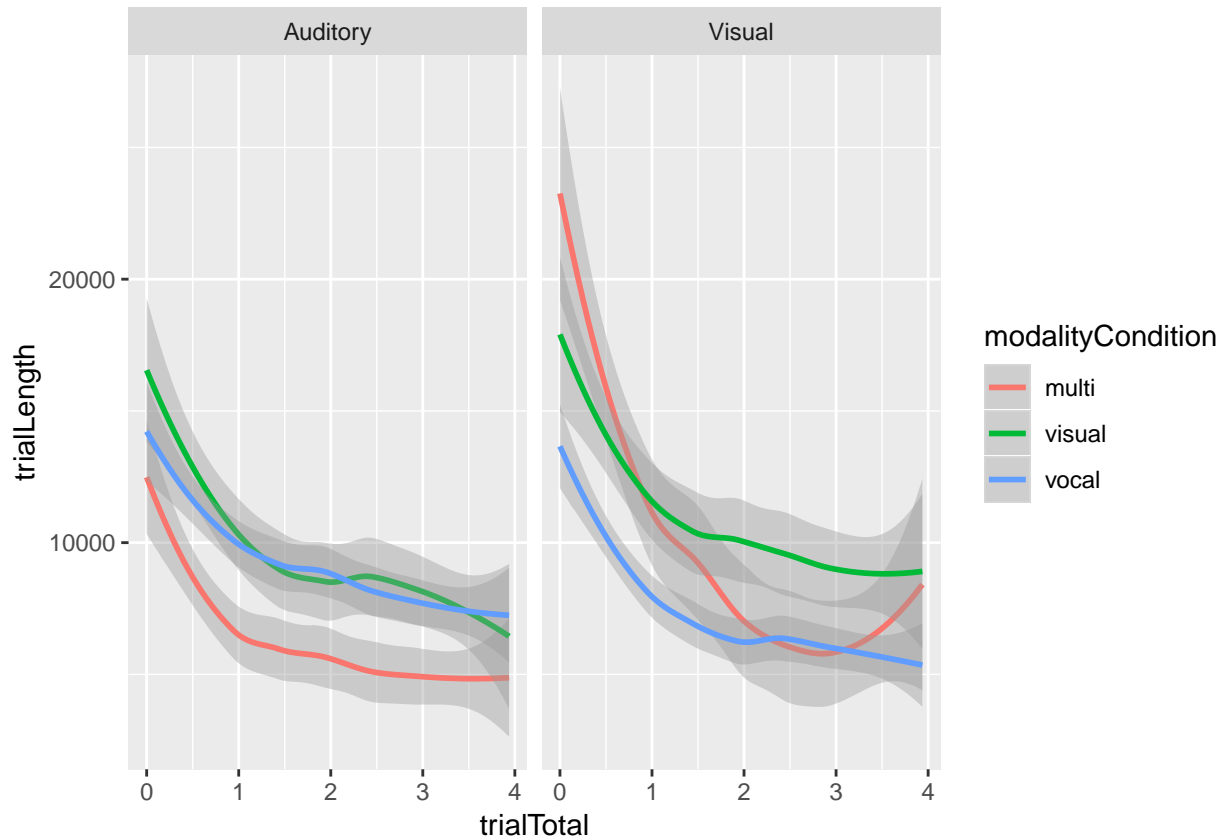
Here is a graph showing the distribution of trial lengths by conditions:

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

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

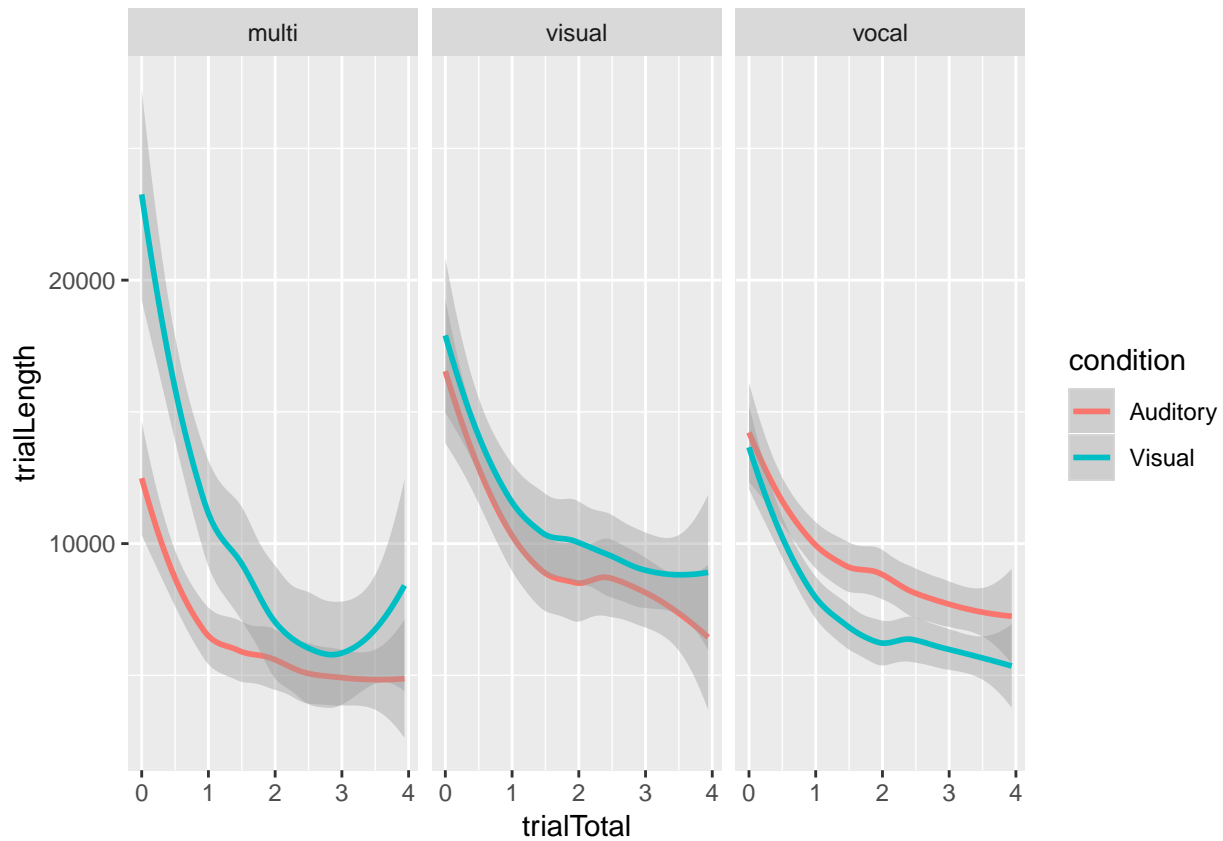
ggplot(d, aes(x=trialTotal, y=trialLength, colour=modalityCondition)) +
  geom_smooth() + facet_grid(.~condition)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



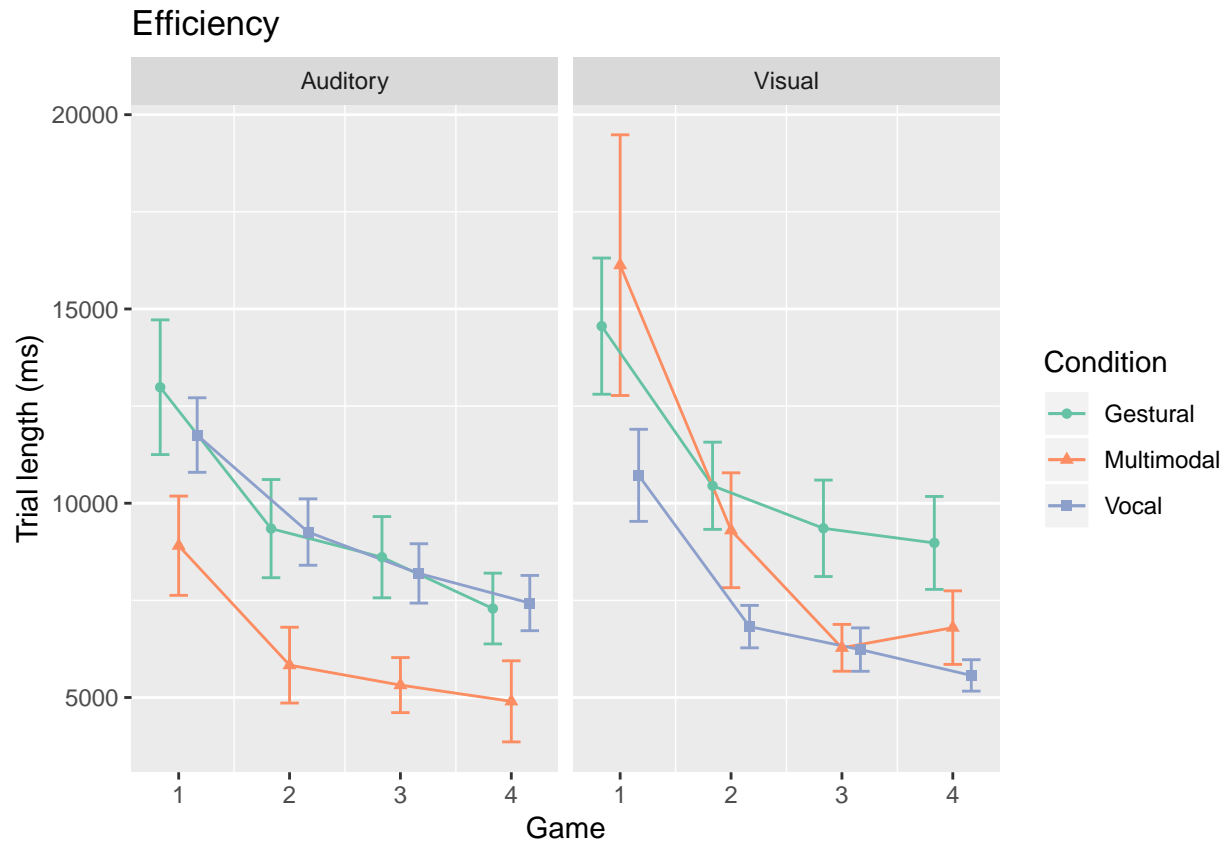
```
ggplot(d, aes(x=trialTotal, y=trialLength, colour=condition)) +
  geom_smooth() + facet_grid(.~modalityCondition)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
pd = position_dodge(width=0.5)
gx1 = ggplot(summary, aes(x=game, y=Efficiency, group=condition, colour=modalityCondition)) +
  geom_errorbar(aes(ymin=lower, ymax=upper, group=modalityCondition), width=0.5, position=pd) +
  facet_grid(. ~ condition) +
  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") +
  theme(panel.grid.major.x = element_blank()) +
  ggtitle("Efficiency") +
  xlab("Game") +
  ylab("Trial length (ms)")
```

gx1

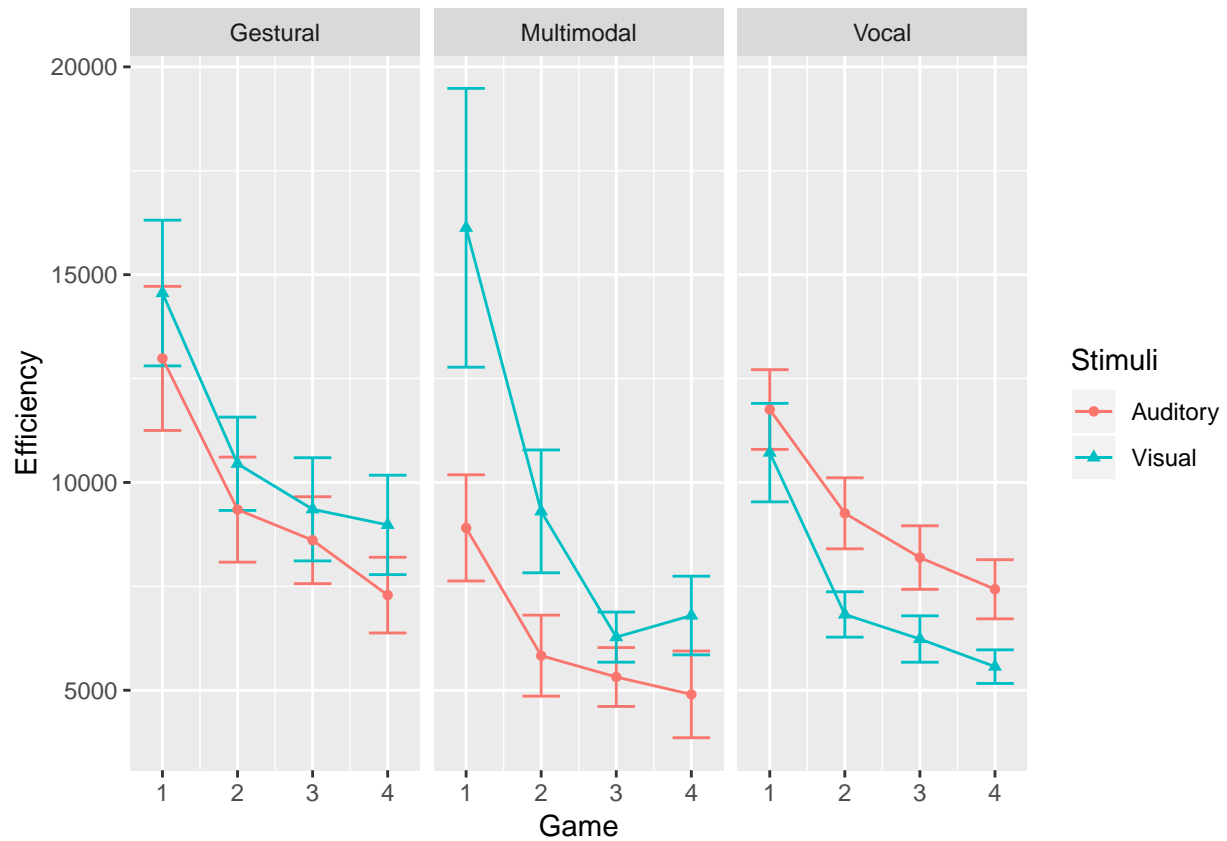


```
pdf("../results/graphs/Efficiency_gg.pdf",
     width = 5, height=3)
gx1
dev.off()

## pdf
## 2

gx2 = ggplot(summary, aes(x=game, y=Efficiency, group=condition, colour=condition, shape=condition)) +
  geom_errorbar(aes(ymin=lower, ymax=upper), width=0.5) +
  facet_grid(. ~ modalityCondition) +
  stat_summary(fun.y="mean", geom="line", aes(group=condition)) +
  geom_point() +
  scale_colour_discrete(name="Stimuli") +
  scale_shape_discrete(name="Stimuli") +
  xlab("Game")

gx2
```



```
pdf("../results/graphs/Efficiency_gg_alt.pdf",
     width = 5, height=3)
gx2
dev.off()
```

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

Average trial time for the whole experiment:

```
mean(d$trialLength)
```

```
## [1] 8795.327
```

```
sd(d$trialLength)
```

```
## [1] 7239.617
```

The distribution of trial times is very skewed:

```
hist(d$trialLength)
```

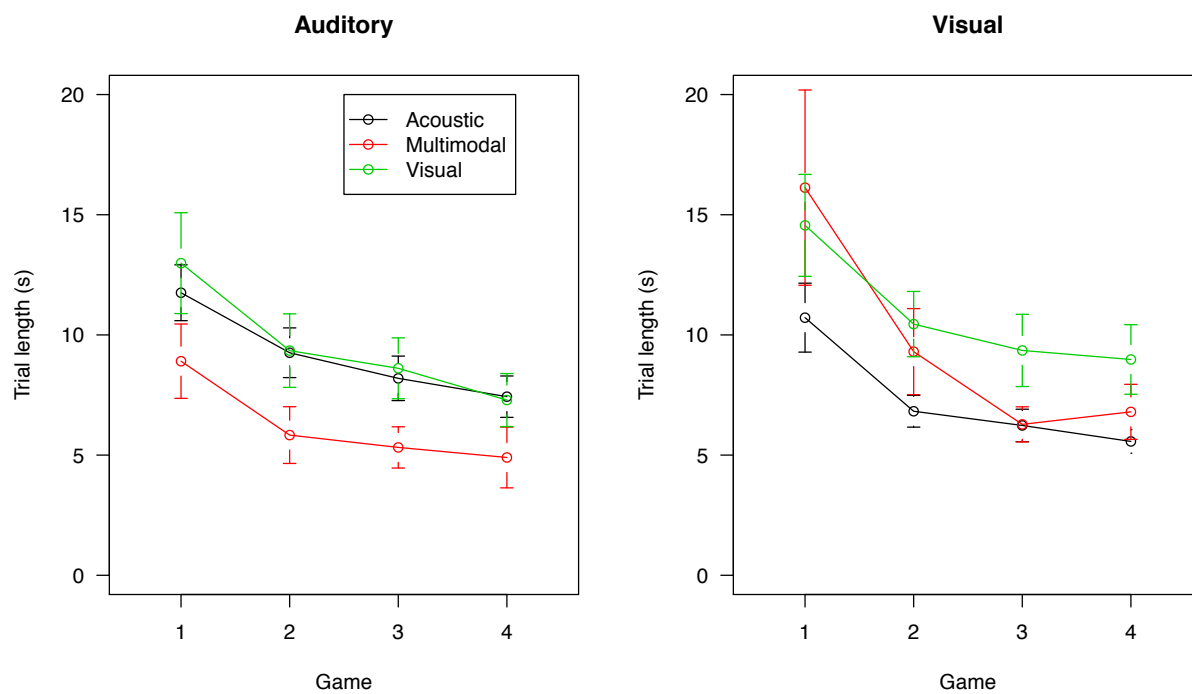
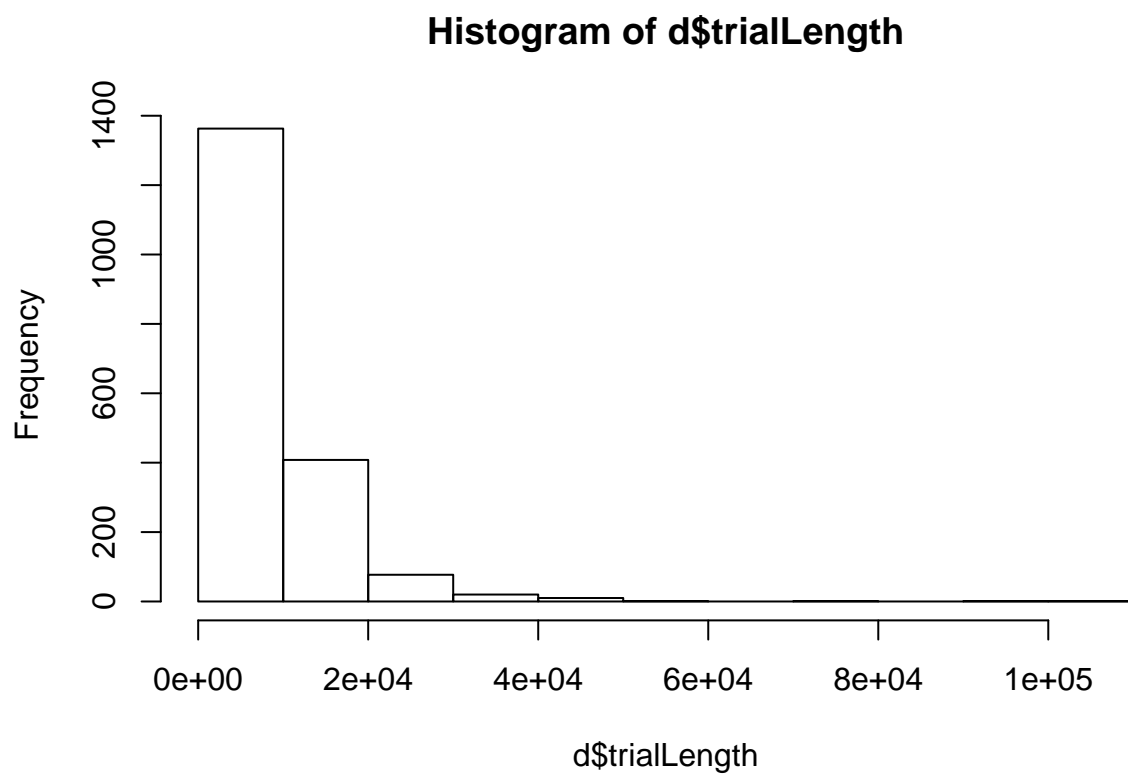


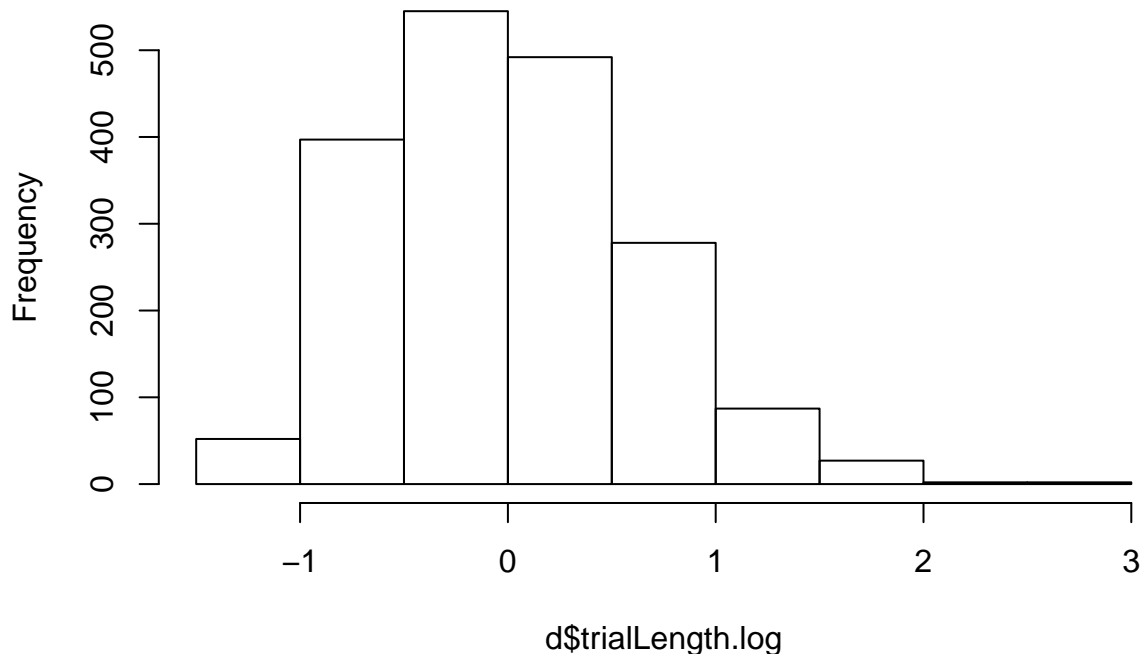
Figure 1: The efficiency of trials in different conditions



So we transform it using a log transform, then center the data.

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

Histogram of d\$trialLength.log



```
# Center the trialTotal variable so intercept reflects after the first game
d$trialTotal = d$trialTotal - 2

matcherResponds.cumulative.mean = mean(d$matcherResponds.cumulative)

d$matcherResponds.cumulative = d$matcherResponds.cumulative - matcherResponds.cumulative.mean

d$matcherResponds = factor(d$matcherResponds)
```

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))])
```

Reorder some levels so that the intercept reflects the most frequent condition.

```
d$incorrect = !d$correct
```

Variable for whether T1 was a multimodal signal.

```
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
```


Mixed models

Make a series of models with random effects for dyad, director (nested within dyad) and item.

Not all random slopes are appropriate. For example, items are used in only one stimulus condition, so a random slope for condition by item is not appropriate. Similarly, each dyad only plays in one modality condition.

It is reasonable to have a random slope for trial by dyad, but this caused unreliable model convergence, so is not included.

The final random slopes were for condition and incorrectness by dyad/player, and modality condition by item.

```
# lme4 version 1.1-21 introduces different default optimisers
# Switch back to original:
ctrl = lmerControl(optimizer = "bobyqa")
# No fixed effects
m0 = lmer(trialLength.log ~ 1 +
          (1 + condition + incorrect | dyadNumber/playerId) +
          (1 + modalityCondition | itemId),
          data=d, REML = FALSE, control = ctrl)
```

boundary (singular) fit: see ?isSingular

Now we add a series of possible confounding factors such as whether the matcher responds. We add the main experimental factors at the end to ensure that they're really contributing to the model over and above the confounds.

```
# Add effect of trial

game = lmer(trialLength.log ~ 1 +
            trialTotal +
            (1 + condition + incorrect | dyadNumber/playerId) +
            (1 + modalityCondition | itemId),
            data=d, REML = FALSE, control = ctrl)

# Add the quadratic effect of trial
gamQuad = lmer(trialLength.log ~ 1 +
               trialTotal + I(trialTotal^2) +
               (1 + condition + incorrect | dyadNumber/playerId) +
               (1 + modalityCondition | itemId),
               data=d, REML = FALSE, control = ctrl)

# Add number of matcher turns
mtchTrn = lmer(trialLength.log ~ 1 +
               trialTotal + I(trialTotal^2) +
               matcherResponds +
               (1 + condition + incorrect | dyadNumber/playerId) +
               (1 + modalityCondition | itemId),
               data=d, REML = FALSE, control = ctrl)
```

boundary (singular) fit: see ?isSingular

```
tMtchTr = lmer(trialLength.log ~ 1 +
               trialTotal + I(trialTotal^2) +
               matcherResponds +
               matcherResponds.cumulative +
               (1 + condition + incorrect | dyadNumber/playerId) +
```

```

      (1 + modalityCondition|itemId),
      data=d, REML = FALSE, control = ctrl)

## boundary (singular) fit: see ?isSingular
# Add whether the response was incorrect
incor = lmer(trialLength.log ~ 1 +
             trialTotal + I(trialTotal^2) +
             matcherResponds +
             matcherResponds.cumulative +
             incorrect +
             (1 + condition + incorrect |dyadNumber/playerId) +
             (1 + modalityCondition|itemId),
             data=d, REML = FALSE, control = ctrl)

## boundary (singular) fit: see ?isSingular
# Add multimodal signal
multim = lmer(trialLength.log ~ 1 +
              trialTotal + I(trialTotal^2) +
              matcherResponds +
              matcherResponds.cumulative +
              incorrect +
              multimodal +
              (1 + condition + incorrect |dyadNumber/playerId) +
              (1 + modalityCondition|itemId),
              data=d, REML = FALSE, control = ctrl)

## boundary (singular) fit: see ?isSingular
# Add modality condition
modality = lmer(trialLength.log ~ 1 + modalityCondition +
                trialTotal + I(trialTotal^2) +
                matcherResponds +
                matcherResponds.cumulative +
                incorrect +
                multimodal +
                (1 + condition + incorrect |dyadNumber/playerId) +
                (1 + modalityCondition|itemId),
                data=d, REML = FALSE, control = ctrl)

## boundary (singular) fit: see ?isSingular
# Add stimulus condition
cond = lmer(trialLength.log ~ 1 + modalityCondition + condition +
            trialTotal + I(trialTotal^2) +
            matcherResponds +
            matcherResponds.cumulative +
            incorrect +
            multimodal +
            (1 + condition + incorrect |dyadNumber/playerId) +
            (1 + modalityCondition|itemId),
            data=d, REML = FALSE, control = ctrl)

## boundary (singular) fit: see ?isSingular

```

```

# Add interaction between modality and stimulus condition
modXcond = lmer(trialLength.log ~ 1 + modalityCondition*condition +
  trialTotal + I(trialTotal^2) +
  matcherResponds +
  matcherResponds.cumulative +
  incorrect +
  multimodal +
  (1 + condition + incorrect |dyadNumber/playerId) +
  (1 + modalityCondition|itemId),
  data=d, REML = FALSE,control = ctrl)

```

boundary (singular) fit: see ?isSingular

```

# Add interaction between condition and trial
conXgame = lmer(trialLength.log ~ 1 + modalityCondition*condition +
  trialTotal + I(trialTotal^2) +
  condition:trialTotal +
  matcherResponds +
  matcherResponds.cumulative +
  incorrect +
  multimodal +
  (1 + condition + incorrect |dyadNumber/playerId) +
  (1 + modalityCondition|itemId),
  data=d, REML = FALSE,control = ctrl)

```

boundary (singular) fit: see ?isSingular

```

# Add interaction between modality and trial
modXgame = lmer(trialLength.log ~ 1 + modalityCondition*condition +
  trialTotal + I(trialTotal^2) +
  condition:trialTotal + modalityCondition:trialTotal +
  matcherResponds +
  matcherResponds.cumulative +
  incorrect +
  multimodal +
  (1 + condition + incorrect |dyadNumber/playerId) +
  (1 + modalityCondition|itemId),
  data=d, REML = FALSE,control = ctrl)

```

boundary (singular) fit: see ?isSingular

```

# Add 3-way interaction
moXcoXga = lmer(trialLength.log ~ 1 + modalityCondition*condition*trialTotal +
  I(trialTotal^2) +
  matcherResponds +
  incorrect +
  multimodal +
  (1 + condition + incorrect |dyadNumber/playerId) +
  (1 + modalityCondition|itemId),
  data=d, REML = FALSE,control = ctrl)

```

boundary (singular) fit: see ?isSingular

Interactions

```

# interaction between turns and modality
nTurnXmo = lmer(trialLength.log ~ 1 + modalityCondition*condition*trialTotal +

```

```

      I(trialTotal^2) +
      matcherResponds + matcherResponds:modalityCondition +
      matcherResponds.cumulative +
      incorrect +
      multimodal +
      (1 + condition + incorrect |dyadNumber/playerId) +
      (1 + modalityCondition|itemId),
      data=d, REML = FALSE,control = ctrl)

## boundary (singular) fit: see ?isSingular

nTurnXco = lmer(trialLength.log ~ 1 + modalityCondition*condition*trialTotal +
      I(trialTotal^2) +
      matcherResponds + matcherResponds:modalityCondition +
      matcherResponds:condition +
      matcherResponds.cumulative +
      incorrect +
      multimodal +
      (1 + condition + incorrect |dyadNumber/playerId) +
      (1 + modalityCondition|itemId),
      data=d, REML = FALSE,control = ctrl)

## boundary (singular) fit: see ?isSingular
# Turn x modality x condition
# Note that the acoustic modality had hardly any matcher turns,
# so the factor is dropped

tuXmoXco = lmer(trialLength.log ~ 1 + modalityCondition*condition*trialTotal +
      I(trialTotal^2) +
      matcherResponds*modalityCondition*condition +
      matcherResponds.cumulative +
      incorrect +
      multimodal +
      (1 + condition + incorrect |dyadNumber/playerId) +
      (1 + modalityCondition|itemId),
      data=d, REML = FALSE,control = ctrl)

## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## boundary (singular) fit: see ?isSingular
# Add the interaction between modality and incorrectness
moXincor = lmer(trialLength.log ~ 1 + modalityCondition*condition*trialTotal +
      I(trialTotal^2) +
      matcherResponds*modalityCondition*condition +
      matcherResponds.cumulative +
      incorrect + incorrect:modalityCondition +
      multimodal +
      (1 + condition + incorrect |dyadNumber/playerId) +
      (1 + modalityCondition|itemId),
      data=d, REML = FALSE,control = ctrl)

## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## boundary (singular) fit: see ?isSingular
# Add the interaction between condition and incorrectness
coXincor = lmer(trialLength.log ~ 1 + modalityCondition*condition*trialTotal +

```

```

I(trialTotal^2) +
matcherResponds*modalityCondition*condition +
matcherResponds.cumulative +
incorrect + incorrect:modalityCondition + incorrect:condition +
multimodal +
(1 + condition + incorrect |dyadNumber/playerId) +
(1 + modalityCondition|itemId),
data=d, REML = FALSE,control = ctrl)

```

```

## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## boundary (singular) fit: see ?isSingular

```

```

# Add the three-way interaction between condition, modality and incorrectness
coXmoXin = lmer(trialLength.log ~ 1 + modalityCondition*condition*trialTotal +
I(trialTotal^2) +
matcherResponds*modalityCondition*condition +
matcherResponds.cumulative +
incorrect *modalityCondition*condition +
multimodal +
(1 + condition + incorrect |dyadNumber/playerId) +
(1 + modalityCondition|itemId),
data=d, REML = FALSE,control = ctrl)

```

```

## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## boundary (singular) fit: see ?isSingular

```

```

# Interaction between multimodality and condition
multiXco = lmer(trialLength.log ~ 1 + modalityCondition*condition*trialTotal +
I(trialTotal^2) +
matcherResponds*modalityCondition*condition +
matcherResponds.cumulative +
incorrect *modalityCondition*condition +
multimodal + multimodal:condition +
(1 + condition + incorrect |dyadNumber/playerId) +
(1 + modalityCondition|itemId),
data=d, REML = FALSE,control = ctrl)

```

```

## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## boundary (singular) fit: see ?isSingular

```

```

# Add interaction between quadratic effect of trial and modality
modXgamQ = lmer(trialLength.log ~ 1 + modalityCondition*condition*trialTotal +
I(trialTotal^2) +(modalityCondition:I(trialTotal^2)) +
matcherResponds*modalityCondition*condition +
matcherResponds.cumulative +
incorrect *modalityCondition*condition +
multimodal + multimodal:condition +
(1 + condition + incorrect |dyadNumber/playerId) +
(1 + modalityCondition|itemId),
data=d, REML = FALSE,control = ctrl)

```

```

## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## boundary (singular) fit: see ?isSingular

```

Interactions with matcher turns

```

tMaTxMod = lmer(trialLength.log ~ 1 + modalityCondition*condition*trialTotal +
  I(trialTotal^2) +(modalityCondition:I(trialTotal^2)) +
  matcherResponds*modalityCondition*condition +
  matcherResponds.cumulative +
  matcherResponds.cumulative:modalityCondition +
  incorrect *modalityCondition*condition +
  multimodal + multimodal:condition +
  (1 + condition + incorrect |dyadNumber/playerId) +
  (1 + modalityCondition|itemId),
  data=d, REML = FALSE, control = ctrl)

## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## boundary (singular) fit: see ?isSingular

Check block has no effect

# Add block order
block = lmer(trialLength.log ~ 1 + modalityCondition*condition*trialTotal +
  I(trialTotal^2) +(modalityCondition:I(trialTotal^2)) +
  matcherResponds*modalityCondition*condition +
  matcherResponds.cumulative +
  matcherResponds.cumulative:modalityCondition +
  incorrect *modalityCondition*condition +
  multimodal + multimodal:condition +
  matcherResponds +
  firstBlock +
  (1 + condition + incorrect |dyadNumber/playerId) +
  (1 + modalityCondition|itemId),
  data=d, REML = TRUE, control = ctrl)

## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## boundary (singular) fit: see ?isSingular

# Last model is REML to get estimates

# Add interaction between block order and modality
blocXmod = lmer(trialLength.log ~ 1 + modalityCondition*condition*trialTotal +
  I(trialTotal^2) +(modalityCondition:I(trialTotal^2)) +
  matcherResponds*modalityCondition*condition +
  matcherResponds.cumulative +
  matcherResponds.cumulative:modalityCondition +
  incorrect *modalityCondition*condition +
  multimodal + multimodal:condition +
  matcherResponds +
  firstBlock*modalityCondition +
  (1 + condition + incorrect |dyadNumber/playerId) +
  (1 + modalityCondition|itemId),
  data=d, REML = TRUE)

## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## boundary (singular) fit: see ?isSingular

```

Results

Compare the fit of the models:

```
modelComparison = anova(m0,modality,cond,game,modXcond,conXgame, modXgame,
  moXcoXga,mtchTrn,tMtchTr,tMaTxMod,nTurnXmo,nTurnXco,tuXmoXco,
  incor,moXincor,coXincor,coXmoXin,
  multim,multiXco,
  gamQuad,modXgamQ,
  block, blocXmod)
```

```
## refitting model(s) with ML (instead of REML)
```

```
attributes(modelComparison)$heading = ""
modelComparison
```

```
##
##          Df      AIC      BIC    logLik deviance    Chisq Chi Df Pr(>Chisq)
## m0        20 2686.0 2796.8 -1323.01   2646.0
## game      21 2287.7 2404.1 -1122.86   2245.7 400.3001      1 < 2.2e-16 ***
## gamQuad   22 2211.0 2332.9 -1083.51   2167.0  78.7027      1 < 2.2e-16 ***
## mtchTrn   23 1721.6 1849.1  -837.82   1675.6 491.3671      1 < 2.2e-16 ***
## tMtchTr   24 1723.6 1856.5  -837.79   1675.6   0.0676      1 0.7948416
## incor     25 1712.3 1850.8  -831.16   1662.3  13.2481      1 0.0002729 ***
## multim    26 1712.8 1856.9  -830.41   1660.8   1.5030      1 0.2202137
## modality  28 1716.2 1871.3  -830.10   1660.2   0.6327      2 0.7287886
## cond      29 1717.7 1878.3  -829.83   1659.7   0.5376      1 0.4634404
## modXcond  31 1706.2 1877.9  -822.10   1644.2  15.4611      2 0.0004392 ***
## conXgame  32 1708.0 1885.3  -822.00   1644.0   0.1885      1 0.6641504
## modXgame  34 1701.5 1889.9  -816.75   1633.5  10.5073      2 0.0052284 **
## moXcoXga  35 1702.9 1896.8  -816.44   1632.9   0.6220      1 0.4303010
## nTurnXmo  38 1706.0 1916.5  -814.98   1630.0   2.9249      3 0.4033423
## nTurnXco  39 1707.8 1923.9  -814.90   1629.8   0.1425      1 0.7058457
## tuXmoXco  40 1709.0 1930.6  -814.48   1629.0   0.8569      1 0.3545994
## moXincor  42 1707.8 1940.5  -811.89   1623.8   5.1708      2 0.0753659 .
## coXincor  43 1709.7 1947.9  -811.86   1623.7   0.0630      1 0.8017564
## coXmoXin  45 1711.5 1960.8  -810.77   1621.5   2.1705      2 0.3378165
## multiXco  46 1713.2 1968.0  -810.58   1621.2   0.3966      1 0.5288414
## modXgamQ  48 1710.3 1976.3  -807.17   1614.3   6.8134      2 0.0331502 *
## tMaTxMod  50 1712.5 1989.5  -806.24   1612.5   1.8515      2 0.3962261
## block     51 1714.1 1996.6  -806.03   1612.1   0.4164      1 0.5187396
## blocXmod  53 1718.4 2012.0  -806.18   1612.4   0.0000      2 1.0000000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Pick final model for estimates:

```
finalModel = block
```

Final model estimates:

```
summary(finalModel)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## trialLength.log ~ 1 + modalityCondition * condition * trialTotal +
##          I(trialTotal^2) + (modalityCondition:I(trialTotal^2)) + matcherResponds *
```

```

##      modalityCondition * condition + matcherResponds.cumulative +
##      matcherResponds.cumulative:modalityCondition + incorrect *
##      modalityCondition * condition + multimodal + multimodal:condition +
##      matcherResponds + firstBlock + (1 + condition + incorrect |
##      dyadNumber/playerId) + (1 + modalityCondition | itemId)
## Data: d
## Control: ctrl
##
## REML criterion at convergence: 1744.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.4137 -0.6137 -0.0552  0.5712  5.7083
##
## Random effects:
##      Groups                Name                Variance Std.Dev.  Corr
## playerId:dyadNumber (Intercept)                0.043220 0.20790
##                      conditionVisual            0.028599 0.16911  -0.58
##                      incorrectTRUE              0.015801 0.12570  -0.72  0.16
## itemId              (Intercept)                0.025369 0.15928
##                      modalityConditionvisual    0.002625 0.05123   0.80
##                      modalityConditionvocal     0.012299 0.11090  -0.09  0.52
## dyadNumber          (Intercept)                0.066131 0.25716
##                      conditionVisual            0.023988 0.15488  -0.13
##                      incorrectTRUE              0.001567 0.03959  -0.46 -0.82
## Residual                                0.123399 0.35128
## Number of obs: 1882, groups:
## playerId:dyadNumber, 30; itemId, 16; dyadNumber, 15
##
## Fixed effects:
##
##                                     Estimate
## (Intercept)                        -0.530835
## modalityConditionvisual              0.498797
## modalityConditionvocal              0.378782
## conditionVisual                    0.409872
## trialTotal                        -0.157898
## I(trialTotal^2)                    0.061550
## matcherRespondsTRUE                0.907786
## matcherResponds.cumulative         -0.019613
## incorrectTRUE                      0.268012
## multimodalTRUE                     0.115162
## firstBlockVisual                   -0.079581
## modalityConditionvisual:conditionVisual -0.247559
## modalityConditionvocal:conditionVisual -0.690612
## modalityConditionvisual:trialTotal    0.018677
## modalityConditionvocal:trialTotal     0.008403
## conditionVisual:trialTotal           -0.001521
## modalityConditionvisual:I(trialTotal^2) -0.036022
## modalityConditionvocal:I(trialTotal^2) -0.002917
## modalityConditionvisual:matcherRespondsTRUE -0.008214
## modalityConditionvocal:matcherRespondsTRUE -0.105875
## conditionVisual:matcherRespondsTRUE   0.088654
## modalityConditionvisual:matcherResponds.cumulative 0.021201
## modalityConditionvocal:matcherResponds.cumulative -0.090406

```


## modalityConditionvisual:incorrectTRUE	-0.077329
## modalityConditionvocal:incorrectTRUE	-0.228144
## conditionVisual:incorrectTRUE	0.027018
## conditionVisual:multimodalTRUE	-0.064590
## modalityConditionvisual:conditionVisual:trialTotal	0.014131
## modalityConditionvocal:conditionVisual:trialTotal	-0.016712
## modalityConditionvisual:conditionVisual:matcherRespondsTRUE	-0.104292
## modalityConditionvisual:conditionVisual:incorrectTRUE	-0.129780
## modalityConditionvocal:conditionVisual:incorrectTRUE	0.064822
##	Std. Error
## (Intercept)	0.173737
## modalityConditionvisual	0.197178
## modalityConditionvocal	0.498299
## conditionVisual	0.132324
## trialTotal	0.019476
## I(trialTotal^2)	0.012668
## matcherRespondsTRUE	0.092247
## matcherResponds.cumulative	0.012612
## incorrectTRUE	0.091586
## multimodalTRUE	0.058240
## firstBlockVisual	0.139013
## modalityConditionvisual:conditionVisual	0.143128
## modalityConditionvocal:conditionVisual	0.157212
## modalityConditionvisual:trialTotal	0.028170
## modalityConditionvocal:trialTotal	0.026254
## conditionVisual:trialTotal	0.025965
## modalityConditionvisual:I(trialTotal^2)	0.017465
## modalityConditionvocal:I(trialTotal^2)	0.017385
## modalityConditionvisual:matcherRespondsTRUE	0.120640
## modalityConditionvocal:matcherRespondsTRUE	0.374660
## conditionVisual:matcherRespondsTRUE	0.116446
## modalityConditionvisual:matcherResponds.cumulative	0.016084
## modalityConditionvocal:matcherResponds.cumulative	0.212914
## modalityConditionvisual:incorrectTRUE	0.121223
## modalityConditionvocal:incorrectTRUE	0.125045
## conditionVisual:incorrectTRUE	0.101040
## conditionVisual:multimodalTRUE	0.107341
## modalityConditionvisual:conditionVisual:trialTotal	0.036040
## modalityConditionvocal:conditionVisual:trialTotal	0.035938
## modalityConditionvisual:conditionVisual:matcherRespondsTRUE	0.156611
## modalityConditionvisual:conditionVisual:incorrectTRUE	0.137872
## modalityConditionvocal:conditionVisual:incorrectTRUE	0.133813
##	t value
## (Intercept)	-3.055
## modalityConditionvisual	2.530
## modalityConditionvocal	0.760
## conditionVisual	3.097
## trialTotal	-8.107
## I(trialTotal^2)	4.859
## matcherRespondsTRUE	9.841
## matcherResponds.cumulative	-1.555
## incorrectTRUE	2.926
## multimodalTRUE	1.977
## firstBlockVisual	-0.572

```

## modalityConditionvisual:conditionVisual -1.730
## modalityConditionvocal:conditionVisual -4.393
## modalityConditionvisual:trialTotal 0.663
## modalityConditionvocal:trialTotal 0.320
## conditionVisual:trialTotal -0.059
## modalityConditionvisual:I(trialTotal^2) -2.063
## modalityConditionvocal:I(trialTotal^2) -0.168
## modalityConditionvisual:matcherRespondsTRUE -0.068
## modalityConditionvocal:matcherRespondsTRUE -0.283
## conditionVisual:matcherRespondsTRUE 0.761
## modalityConditionvisual:matcherResponds.cumulative 1.318
## modalityConditionvocal:matcherResponds.cumulative -0.425
## modalityConditionvisual:incorrectTRUE -0.638
## modalityConditionvocal:incorrectTRUE -1.825
## conditionVisual:incorrectTRUE 0.267
## conditionVisual:multimodalTRUE -0.602
## modalityConditionvisual:conditionVisual:trialTotal 0.392
## modalityConditionvocal:conditionVisual:trialTotal -0.465
## modalityConditionvisual:conditionVisual:matcherRespondsTRUE -0.666
## modalityConditionvisual:conditionVisual:incorrectTRUE -0.941
## modalityConditionvocal:conditionVisual:incorrectTRUE 0.484

##
## Correlation matrix not shown by default, as p = 32 > 12.
## Use print(x, correlation=TRUE) or
##      vcov(x)          if you need it

## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## convergence code: 0
## boundary (singular) fit: see ?isSingular

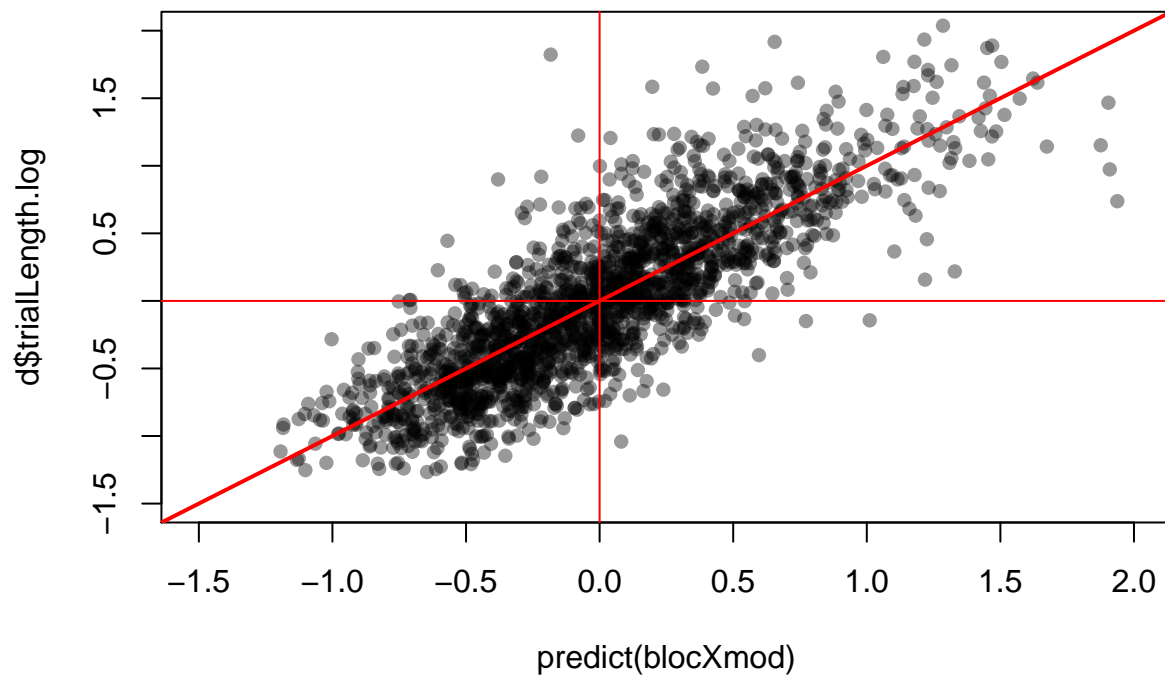
```

Check model predictions. The model predictions are in the right range and direction, fitting linear quite well:

```

plot(predict(blocXmod),d$trialLength.log, pch=16, col=rgb(0,0,0.4),
      ylim=c(-1.5,2),xlim=c(-1.5,2))
abline(a=0,b=1, col=2, lwd=2)
abline(h=0, col=2)
abline(v=0, col=2)

```



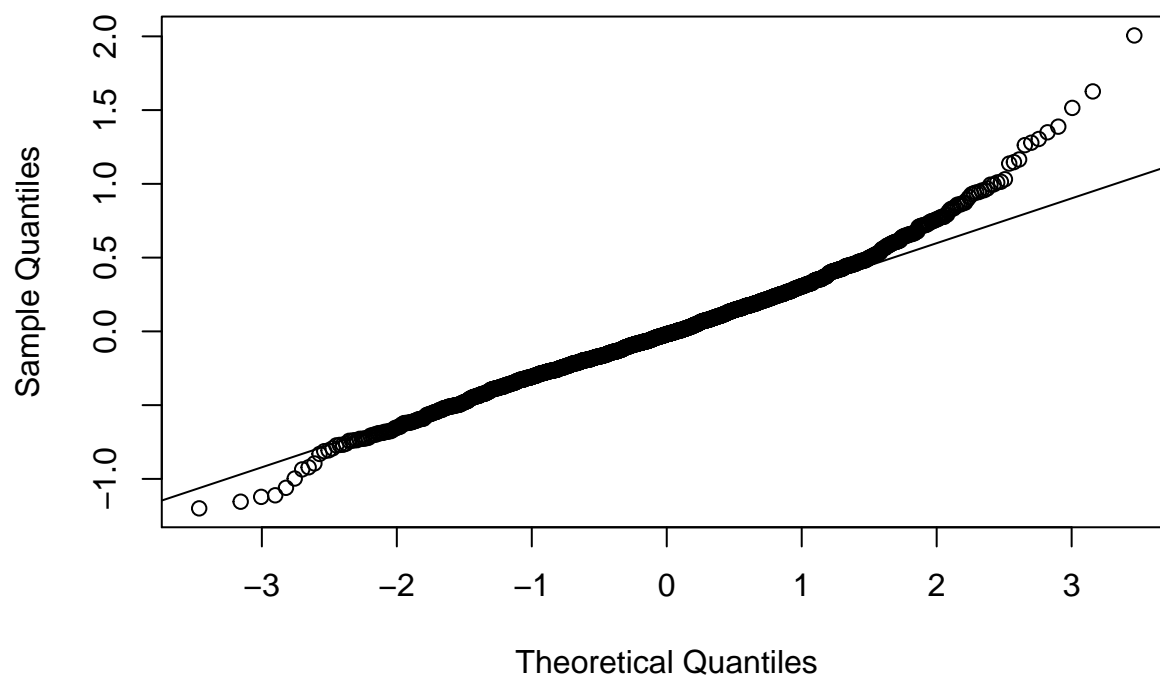
```
cor(predict(finalModel), d$trialLength.log)
```

```
## [1] 0.8336072
```

The residuals are ok, though it tends to do worse at higher values. This is expected from using the log scale.

```
qqnorm(resid(blocXmod))
qqline(resid(blocXmod))
```

Normal Q-Q Plot



Plot the fixed effects

Relabel the effects:

```
feLabels = matrix(c(
  "(Intercept)" , "Intercept" , NA,
  "modalityConditionvisual" , "Visual modality", "modality",
  "modalityConditionvocal" , "Acoustic modality", "modality",
  "conditionVisual" , "Visual stimuli", "cond",
  "trialTotal" , "Game", "game",
  "modalityConditionvisual:conditionVisual" , "Visual modality:Visual stimuli", "modXcond",
  "modalityConditionvocal:conditionVisual" , "Acoustic modality:Visual stimuli", "modXcond",
  "modalityConditionvisual:trialTotal" , "Visual modality:Game", "modXgame",
  "modalityConditionvocal:trialTotal" , "Acoustic modality:Game", "modXgame",
  "conditionVisual:trialTotal" , "Visual stimuli:Game", "conXgame",
  "modalityConditionvisual:conditionVisual:trialTotal", "Visual modality:Visual stimuli:Game", "moXcoXga",
  "modalityConditionvocal:conditionVisual:trialTotal", "Acoustic modality:Visual stimuli:Game", "moXcoXga",
  "incorrectTRUE", "Incorrect", "incor",
  "modalityConditionvisual:incorrectTRUE", "Visual modality:Incorrect", "moXincor",
  "modalityConditionvocal:incorrectTRUE", "Acoustic modality:Incorrect", "moXincor",
  "modalityConditionvisual:I(trialTotal^2)", "Visual modality:Game^2", "modXgamQ",
  "modalityConditionvocal:I(trialTotal^2)", "Acoustic modality:Game^2", "modXgamQ",
  "I(trialTotal^2)", "Game^2", "gamQuad",
  "firstBlockVisual", "Visual stims first", "block",
  "modalityConditionvisual:firstBlockVisual", "Visual modality:Visual stim first", "blocXmod",
  "modalityConditionvocal:firstBlockVisual", "Acoustic modality:Visual stim first", "blocXmod",
  "conditionVisual:incorrectTRUE", "Visual stimuli:incorrect", "coXincor",
  "modalityConditionvisual:conditionVisual:incorrectTRUE", "Visual modality:Visual stimuli:incorrect", "coXincor",
  "modalityConditionvocal:conditionVisual:incorrectTRUE", "Acoustic modality:Visual stimuli:incorrect", "coXincor",
  "modalityConditionvisual:conditionVisual:numberOfTurns", "VisualModality:Visual stim:NumTurns", "tuXmoXco",
  "modalityConditionvisual:conditionVisual:matcherRespondsTRUE", "VisualModality:Visual stim:Matcher Respon",
  "modalityConditionvocal:conditionVisual:numberOfTurns", "Vocal Modality:Visual stim:NumTurns", "tuXmoXco",
  "modalityConditionvocal:conditionVisual:matcherRespondsTRUE", "Vocal Modality:Visual stim:Matcher Respon",
  "conditionVisual:numberOfTurns", "Visual stim:NumTurns", "nTurnXco",
  "conditionVisual:matcherRespondsTRUE", "Visual stim:Matcher Responds", "nTurnXco",
  "modalityConditionvisual:numberOfTurns", "VisualModality:NumTurns", "nTurnXmo",
  "modalityConditionvisual:matcherRespondsTRUE", "VisualModality:Matcher Responds", "nTurnXmo",
  "modalityConditionvocal:numberOfTurns", "Vocal Modality:NumTurns", "nTurnXmo",
  "modalityConditionvocal:matcherRespondsTRUE", "Vocal Modality:Matcher Responds", "nTurnXmo",
  "numberOfTurns", "Number of turns", "nTurns",
  "multimodalTRUE", "Multimodal T1", "multim",
  "conditionVisual:multimodalTRUE", "VisualStim:MultimodalT1", "multiXco",
  "matcherRespondsTRUE", "Matcher Responds", 'mtchTrn',
  "matcherResponds.cumulative", "Total interaction", "tMtchTr",
  "modalityConditionvisual:matcherResponds.cumulative", "Total interaction:Visual Modality", "tMaTxMod",
  "modalityConditionvocal:matcherResponds.cumulative", "Total interaction:Vocal Modality", "tMaTxMod"
), ncol=3, byrow = T)

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])

cols= c("black",'red')
sig.data$pointCol = cols[1]
sig.data$pointCol[!is.na(sig.data$sig)] =
  cols[1 + (sig.data$sig[!is.na(sig.data$sig)] < 0.05)]
# Mark marginal effects
#sig.data$pointCol[!is.na(sig.data$sig) &
#
#
#sig.data$sig < 0.1 &
#
sig.data$sig >=0.05] = "orange"

sig.data$fade = sig.data$sig > 0.05

```

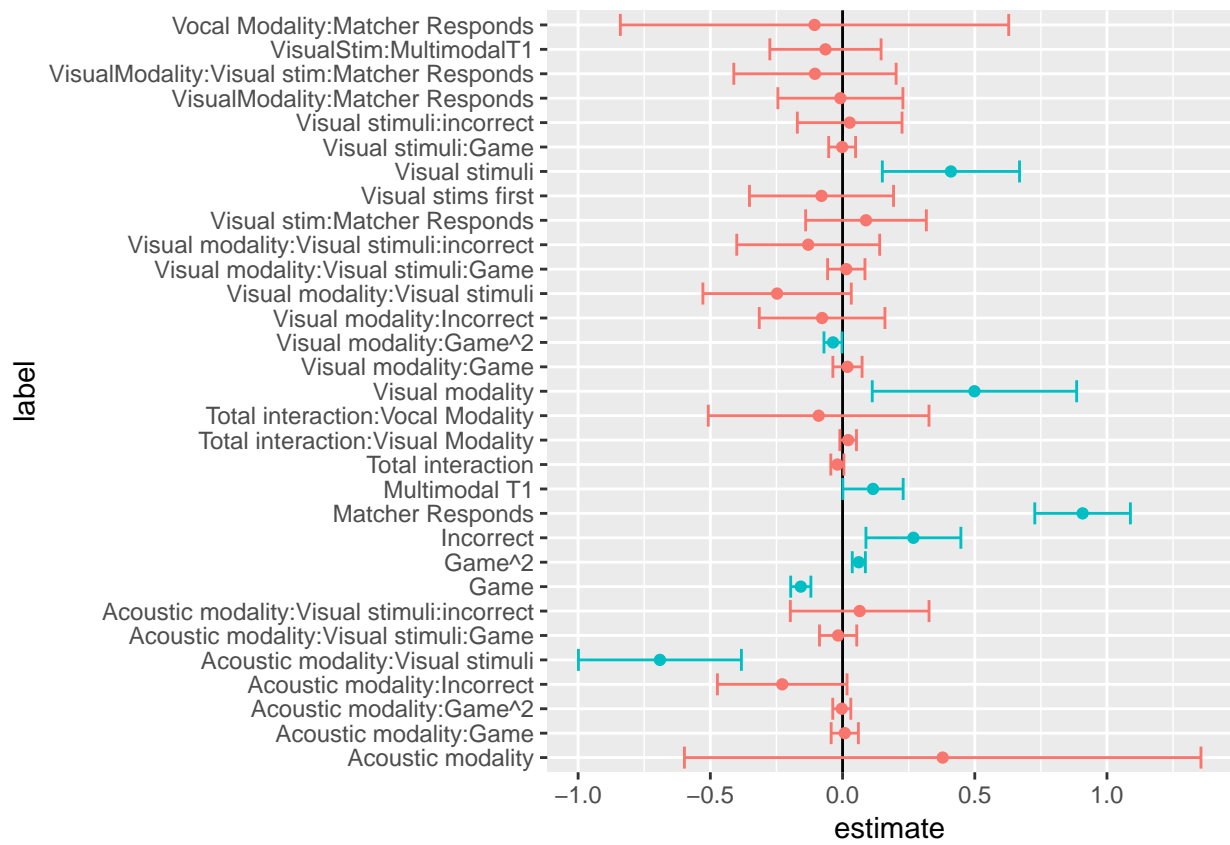
Plot the strength of the fixed effects:

```

x = get_model_data(finalModel, 'est')

## Computing p-values via Wald-statistics approximation (treating t as Wald z).
x$label = feLabels2[2:length(feLabels2)]
x$col = c("sig","nonsig")[1+(x$p.value>=0.05)]
ggplot(x, aes(y=estimate,x=label,colour=col)) +
  geom_hline(yintercept = 0) +
  geom_point() +
  geom_errorbar(aes(ymin=x$conf.low,ymax=x$conf.high)) +
  coord_flip() +
  theme(legend.position = 'none')

```

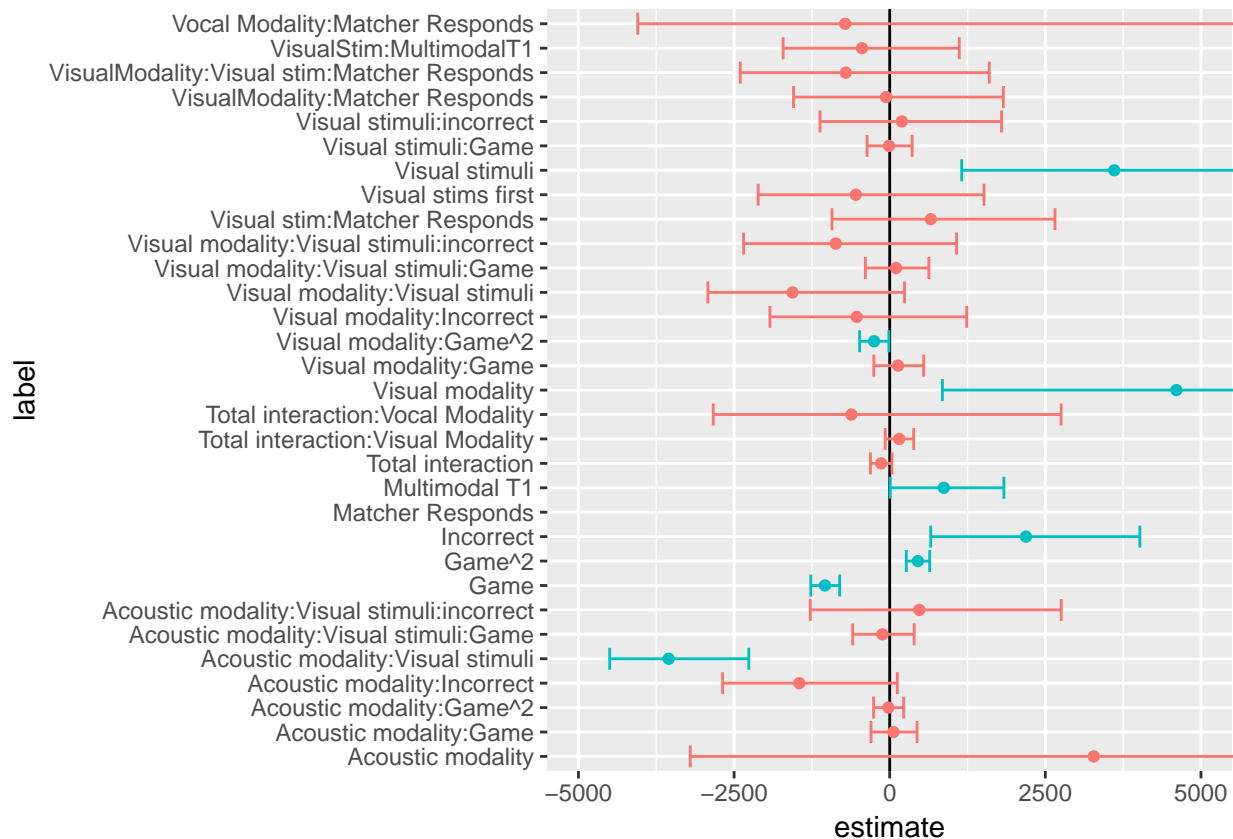


Attempt plot with axes in milliseconds.

```
convertEst = function(X){
  exp(meanLogTrialLength+X) - exp(meanLogTrialLength)
}

x$estimate = convertEst(x$estimate)
x$conf.low = convertEst(x$conf.low)
x$conf.high = convertEst(x$conf.high)

ggplot(x, aes(y=estimate,x=label,colour=col)) +
  geom_hline(yintercept = 0) +
  geom_point() +
  geom_errorbar(aes(ymin=x$conf.low,ymax=x$conf.high)) +
  theme(legend.position = 'none') +
  coord_flip(ylim = c(-5000,5000))
```



for every 10 trials where a matcher responded, subsequent trials were shorter by:

```
noInteraction = convertEst(
  fixef(finalModel)["(Intercept)"]
)
tenResponses = convertEst(
  fixef(finalModel)["(Intercept)"] +
  (10 * fixef(finalModel)["matcherResponds.cumulative"])
)
noInteraction - tenResponses

## (Intercept)
```

745.7498

Table for paper

```
outdata = x[,c("term", "label", "estimate", "conf.low", 'conf.high')]

outdata$estimate = base::round(outdata$estimate)
outdata$conf.low = base::round(outdata$conf.low)
outdata$conf.high = base::round(outdata$conf.high)

xd = as.data.frame(summary(finalModel)$coef)
outdata$wald.t = xd[match(outdata$term, rownames(xd)), 't value']

outdata = cbind(outdata,
  modelComparison[feModel[2:length(feModel)], c("logLik", "Chisq", "Pr(>Chisq)")]
outdata$estimate = paste(
  c("", "+") [1+(outdata$estimate>0)],
  as.character(outdata$estimate), sep='')

outdata = outdata[,c("label", "estimate", "conf.low",
  "conf.high", "wald.t", "Chisq",
  "Pr(>Chisq)")]

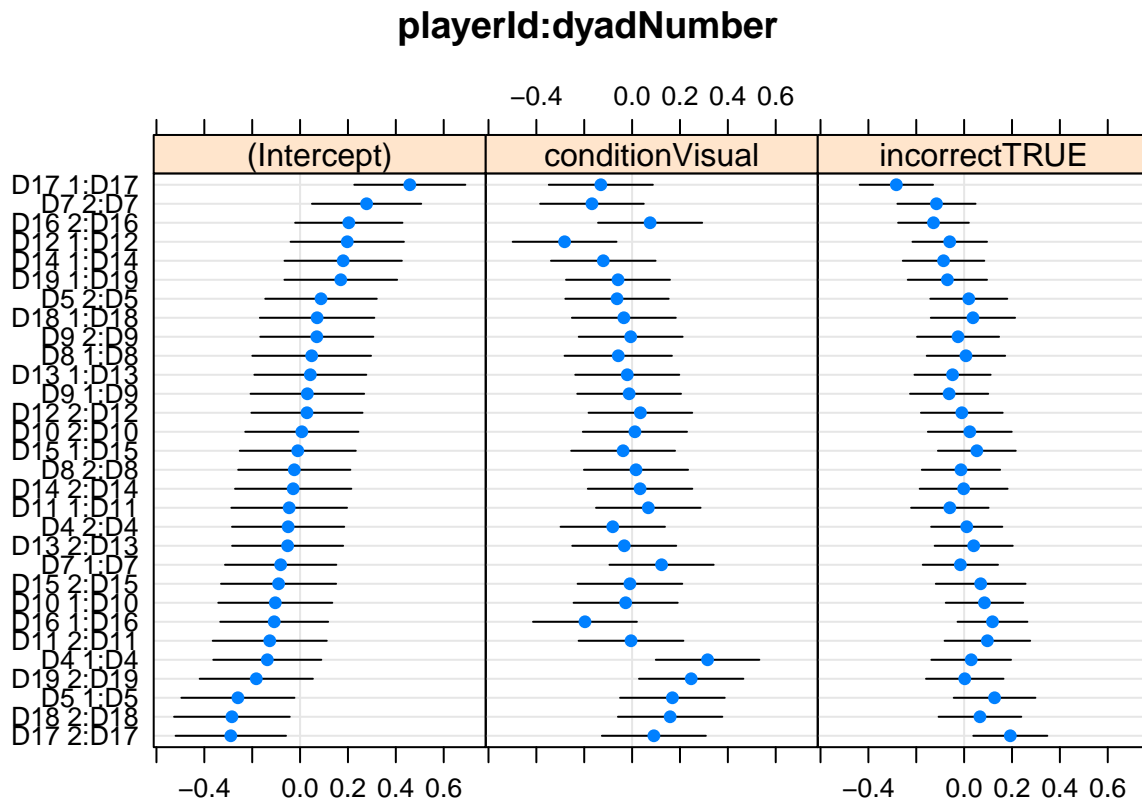
finalRes = outdata
write.csv(finalRes, file="../../results/tables/Efficiency_FixedEffects.csv")
```

Random effects

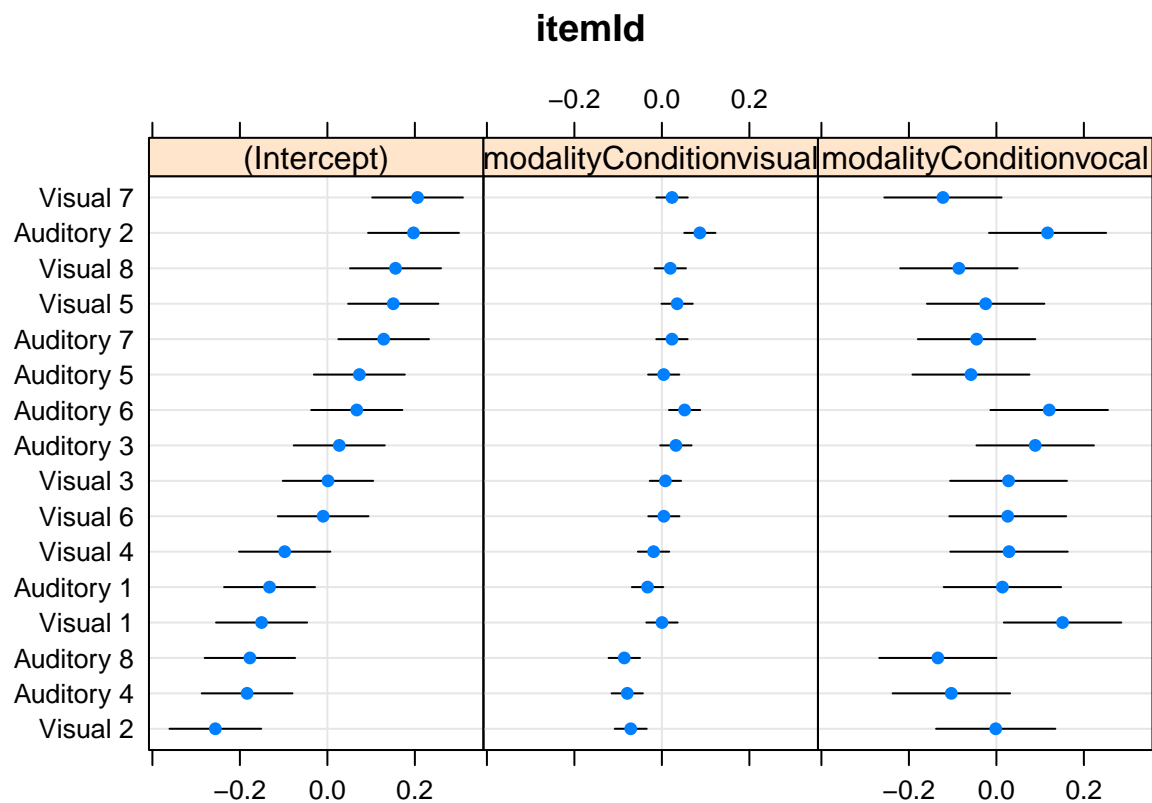
There is a reasonable amount of variation in the random effects, suggesting that dyads and players differ. This justifies the use of mixed effects modelling.

```
dotplot(ranef(finalModel))

## $`playerId:dyadNumber`
```

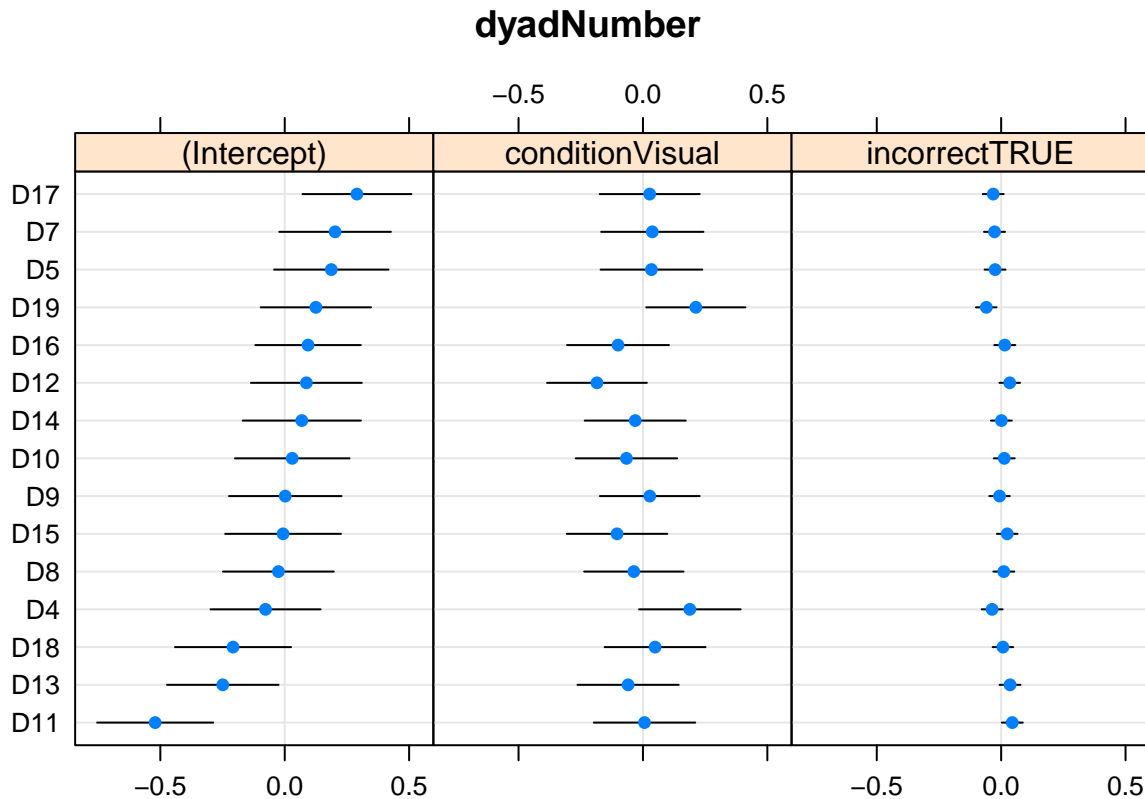


```
##
## $ItemId
```



```
##
```

```
## $dyadNumber
```



Relevel factors to see other comparisons

```
d2 = d
d2$condition = relevel(d2$condition, "Visual")
fm2 = update(finalModel, data=d2)

## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## boundary (singular) fit: see ?isSingular
summary(fm2)

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## trialLength.log ~ 1 + modalityCondition * condition * trialTotal +
## I(trialTotal^2) + (modalityCondition:I(trialTotal^2)) + matcherResponds *
## modalityCondition * condition + matcherResponds.cumulative +
## matcherResponds.cumulative:modalityCondition + incorrect *
## modalityCondition * condition + multimodal + multimodal:condition +
## matcherResponds + firstBlock + (1 + condition + incorrect |
## dyadNumber/playerId) + (1 + modalityCondition | itemId)
## Data: d2
## Control: ctrl
##
## REML criterion at convergence: 1744.1
##
## Scaled residuals:
```

```

##      Min      1Q  Median      3Q      Max
## -3.4137 -0.6137 -0.0552  0.5712  5.7083
##
## Random effects:
##      Groups              Name              Variance Std.Dev. Corr
## playerId:dyadNumber (Intercept)          0.030839 0.17561
##                      conditionAuditory    0.028599 0.16911  -0.27
##                      incorrectTRUE         0.015801 0.12570  -0.70 -0.16
## itemId              (Intercept)          0.025369 0.15928
##                      modalityConditionvisual 0.002625 0.05123   0.80
##                      modalityConditionvocal  0.012299 0.11090  -0.09  0.52
## dyadNumber          (Intercept)          0.079952 0.28276
##                      conditionAuditory    0.023988 0.15488  -0.43
##                      incorrectTRUE         0.001567 0.03959  -0.87  0.82
## Residual                                0.123399 0.35128
## Number of obs: 1882, groups:
## playerId:dyadNumber, 30; itemId, 16; dyadNumber, 15
##
## Fixed effects:
##
##                                     Estimate
## (Intercept)                        -0.120964
## modalityConditionvisual              0.251238
## modalityConditionvocal              -0.311830
## conditionAuditory                   -0.409872
## trialTotal                          -0.159418
## I(trialTotal^2)                     0.061550
## matcherRespondsTRUE                 0.996439
## matcherResponds.cumulative          -0.019613
## incorrectTRUE                       0.295030
## multimodalTRUE                      0.050572
## firstBlockVisual                    -0.079581
## modalityConditionvisual:conditionAuditory 0.247559
## modalityConditionvocal:conditionAuditory 0.690612
## modalityConditionvisual:trialTotal     0.032808
## modalityConditionvocal:trialTotal     -0.008309
## conditionAuditory:trialTotal          0.001521
## modalityConditionvisual:I(trialTotal^2) -0.036022
## modalityConditionvocal:I(trialTotal^2) -0.002917
## modalityConditionvisual:matcherRespondsTRUE -0.112507
## modalityConditionvocal:matcherRespondsTRUE -0.105875
## conditionAuditory:matcherRespondsTRUE -0.088654
## modalityConditionvisual:matcherResponds.cumulative 0.021201
## modalityConditionvocal:matcherResponds.cumulative -0.090406
## modalityConditionvisual:incorrectTRUE -0.207109
## modalityConditionvocal:incorrectTRUE -0.163323
## conditionAuditory:incorrectTRUE       -0.027018
## conditionAuditory:multimodalTRUE      0.064590
## modalityConditionvisual:conditionAuditory:trialTotal -0.014131
## modalityConditionvocal:conditionAuditory:trialTotal 0.016712
## modalityConditionvisual:conditionAuditory:matcherRespondsTRUE 0.104292
## modalityConditionvisual:conditionAuditory:incorrectTRUE 0.129780
## modalityConditionvocal:conditionAuditory:incorrectTRUE -0.064822
##                                     Std. Error
## (Intercept)                        0.176530

```

## modalityConditionvisual	0.202962
## modalityConditionvocal	0.539640
## conditionAuditory	0.132324
## trialTotal	0.022960
## I(trialTotal^2)	0.012668
## matcherRespondsTRUE	0.072111
## matcherResponds.cumulative	0.012612
## incorrectTRUE	0.076437
## multimodalTRUE	0.089998
## firstBlockVisual	0.139012
## modalityConditionvisual:conditionAuditory	0.143128
## modalityConditionvocal:conditionAuditory	0.157212
## modalityConditionvisual:trialTotal	0.031605
## modalityConditionvocal:trialTotal	0.028902
## conditionAuditory:trialTotal	0.025965
## modalityConditionvisual:I(trialTotal^2)	0.017465
## modalityConditionvocal:I(trialTotal^2)	0.017385
## modalityConditionvisual:matcherRespondsTRUE	0.101445
## modalityConditionvocal:matcherRespondsTRUE	0.374660
## conditionAuditory:matcherRespondsTRUE	0.116446
## modalityConditionvisual:matcherResponds.cumulative	0.016084
## modalityConditionvocal:matcherResponds.cumulative	0.212914
## modalityConditionvisual:incorrectTRUE	0.109659
## modalityConditionvocal:incorrectTRUE	0.100925
## conditionAuditory:incorrectTRUE	0.101040
## conditionAuditory:multimodalTRUE	0.107341
## modalityConditionvisual:conditionAuditory:trialTotal	0.036040
## modalityConditionvocal:conditionAuditory:trialTotal	0.035938
## modalityConditionvisual:conditionAuditory:matcherRespondsTRUE	0.156611
## modalityConditionvisual:conditionAuditory:incorrectTRUE	0.137873
## modalityConditionvocal:conditionAuditory:incorrectTRUE	0.133813
##	t value
## (Intercept)	-0.685
## modalityConditionvisual	1.238
## modalityConditionvocal	-0.578
## conditionAuditory	-3.097
## trialTotal	-6.943
## I(trialTotal^2)	4.859
## matcherRespondsTRUE	13.818
## matcherResponds.cumulative	-1.555
## incorrectTRUE	3.860
## multimodalTRUE	0.562
## firstBlockVisual	-0.572
## modalityConditionvisual:conditionAuditory	1.730
## modalityConditionvocal:conditionAuditory	4.393
## modalityConditionvisual:trialTotal	1.038
## modalityConditionvocal:trialTotal	-0.287
## conditionAuditory:trialTotal	0.059
## modalityConditionvisual:I(trialTotal^2)	-2.063
## modalityConditionvocal:I(trialTotal^2)	-0.168
## modalityConditionvisual:matcherRespondsTRUE	-1.109
## modalityConditionvocal:matcherRespondsTRUE	-0.283
## conditionAuditory:matcherRespondsTRUE	-0.761
## modalityConditionvisual:matcherResponds.cumulative	1.318

```

## modalityConditionvocal:matcherResponds.cumulative -0.425
## modalityConditionvisual:incorrectTRUE -1.889
## modalityConditionvocal:incorrectTRUE -1.618
## conditionAuditory:incorrectTRUE -0.267
## conditionAuditory:multimodalTRUE 0.602
## modalityConditionvisual:conditionAuditory:trialTotal -0.392
## modalityConditionvocal:conditionAuditory:trialTotal 0.465
## modalityConditionvisual:conditionAuditory:matcherRespondsTRUE 0.666
## modalityConditionvisual:conditionAuditory:incorrectTRUE 0.941
## modalityConditionvocal:conditionAuditory:incorrectTRUE -0.484

##
## Correlation matrix not shown by default, as p = 32 > 12.
## Use print(x, correlation=TRUE) or
##     vcov(x)         if you need it

## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## convergence code: 0
## boundary (singular) fit: see ?isSingular

d2 = d
d2$modalityCondition = relevel(d2$modalityCondition, "visual")
fm2 = update(finalModel, data=d2)

## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## boundary (singular) fit: see ?isSingular

summary(fm2)

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## trialLength.log ~ 1 + modalityCondition * condition * trialTotal +
##   I(trialTotal^2) + (modalityCondition:I(trialTotal^2)) + matcherResponds *
##   modalityCondition * condition + matcherResponds.cumulative +
##   matcherResponds.cumulative:modalityCondition + incorrect *
##   modalityCondition * condition + multimodal + multimodal:condition +
##   matcherResponds + firstBlock + (1 + condition + incorrect |
##   dyadNumber/playerId) + (1 + modalityCondition | itemId)
## Data: d2
## Control: ctrl
##
## REML criterion at convergence: 1744.1
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.4137 -0.6137 -0.0552  0.5712  5.7083
##
## Random effects:
##   Groups                Name                Variance Std.Dev. Corr
##   playerId:dyadNumber (Intercept)          0.043220 0.20790
##                      conditionVisual      0.028599 0.16911 -0.58
##                      incorrectTRUE         0.015801 0.12570 -0.72  0.16
##   itemId                (Intercept)          0.041101 0.20273
##                      modalityConditionmulti 0.002625 0.05123 -0.88
##                      modalityConditionvocal 0.009038 0.09507 -0.41 -0.07

```

```

## dyadNumber          (Intercept)          0.066131 0.25716
##                   conditionVisual        0.023988 0.15488 -0.13
##                   incorrectTRUE          0.001567 0.03959 -0.46 -0.82
## Residual                                0.123399 0.35128
## Number of obs: 1882, groups:
## playerId:dyadNumber, 30; itemId, 16; dyadNumber, 15
##
## Fixed effects:
##
##                                     Estimate
## (Intercept)                        -0.032037
## modalityConditionmulti              -0.498797
## modalityConditionvocal              -0.120017
## conditionVisual                     0.162312
## trialTotal                         -0.139220
## I(trialTotal^2)                     0.025529
## matcherRespondsTRUE                 0.899571
## matcherResponds.cumulative          0.001588
## incorrectTRUE                       0.190682
## multimodalTRUE                      0.115162
## firstBlockVisual                    -0.079582
## modalityConditionmulti:conditionVisual 0.247559
## modalityConditionvocal:conditionVisual -0.443053
## modalityConditionmulti:trialTotal    -0.018677
## modalityConditionvocal:trialTotal    -0.010274
## conditionVisual:trialTotal           0.012610
## modalityConditionmulti:I(trialTotal^2) 0.036022
## modalityConditionvocal:I(trialTotal^2) 0.033105
## modalityConditionmulti:matcherRespondsTRUE 0.008214
## modalityConditionvocal:matcherRespondsTRUE -0.097661
## conditionVisual:matcherRespondsTRUE -0.015639
## modalityConditionmulti:matcherResponds.cumulative -0.021201
## modalityConditionvocal:matcherResponds.cumulative -0.111608
## modalityConditionmulti:incorrectTRUE 0.077329
## modalityConditionvocal:incorrectTRUE -0.150815
## conditionVisual:incorrectTRUE        -0.102762
## conditionVisual:multimodalTRUE       -0.064590
## modalityConditionmulti:conditionVisual:trialTotal -0.014131
## modalityConditionvocal:conditionVisual:trialTotal -0.030843
## modalityConditionmulti:conditionVisual:matcherRespondsTRUE 0.104292
## modalityConditionmulti:conditionVisual:incorrectTRUE 0.129780
## modalityConditionvocal:conditionVisual:incorrectTRUE 0.194602
##                                     Std. Error
## (Intercept)                        0.174969
## modalityConditionmulti              0.197178
## modalityConditionvocal              0.496076
## conditionVisual                     0.137681
## trialTotal                         0.020395
## I(trialTotal^2)                     0.012021
## matcherRespondsTRUE                 0.077698
## matcherResponds.cumulative          0.010028
## incorrectTRUE                       0.080744
## multimodalTRUE                      0.058240
## firstBlockVisual                    0.139012
## modalityConditionmulti:conditionVisual 0.143128

```

## modalityConditionvocal:conditionVisual	0.146378
## modalityConditionmulti:trialTotal	0.028170
## modalityConditionvocal:trialTotal	0.026928
## conditionVisual:trialTotal	0.025117
## modalityConditionmulti:I(trialTotal^2)	0.017465
## modalityConditionvocal:I(trialTotal^2)	0.016917
## modalityConditionmulti:matcherRespondsTRUE	0.120640
## modalityConditionvocal:matcherRespondsTRUE	0.371386
## conditionVisual:matcherRespondsTRUE	0.104926
## modalityConditionmulti:matcherResponds.cumulative	0.016084
## modalityConditionvocal:matcherResponds.cumulative	0.212751
## modalityConditionmulti:incorrectTRUE	0.121223
## modalityConditionvocal:incorrectTRUE	0.117060
## conditionVisual:incorrectTRUE	0.095733
## conditionVisual:multimodalTRUE	0.107341
## modalityConditionmulti:conditionVisual:trialTotal	0.036040
## modalityConditionvocal:conditionVisual:trialTotal	0.035299
## modalityConditionmulti:conditionVisual:matcherRespondsTRUE	0.156611
## modalityConditionmulti:conditionVisual:incorrectTRUE	0.137872
## modalityConditionvocal:conditionVisual:incorrectTRUE	0.129480
##	t value
## (Intercept)	-0.183
## modalityConditionmulti	-2.530
## modalityConditionvocal	-0.242
## conditionVisual	1.179
## trialTotal	-6.826
## I(trialTotal^2)	2.124
## matcherRespondsTRUE	11.578
## matcherResponds.cumulative	0.158
## incorrectTRUE	2.362
## multimodalTRUE	1.977
## firstBlockVisual	-0.572
## modalityConditionmulti:conditionVisual	1.730
## modalityConditionvocal:conditionVisual	-3.027
## modalityConditionmulti:trialTotal	-0.663
## modalityConditionvocal:trialTotal	-0.382
## conditionVisual:trialTotal	0.502
## modalityConditionmulti:I(trialTotal^2)	2.063
## modalityConditionvocal:I(trialTotal^2)	1.957
## modalityConditionmulti:matcherRespondsTRUE	0.068
## modalityConditionvocal:matcherRespondsTRUE	-0.263
## conditionVisual:matcherRespondsTRUE	-0.149
## modalityConditionmulti:matcherResponds.cumulative	-1.318
## modalityConditionvocal:matcherResponds.cumulative	-0.525
## modalityConditionmulti:incorrectTRUE	0.638
## modalityConditionvocal:incorrectTRUE	-1.288
## conditionVisual:incorrectTRUE	-1.073
## conditionVisual:multimodalTRUE	-0.602
## modalityConditionmulti:conditionVisual:trialTotal	-0.392
## modalityConditionvocal:conditionVisual:trialTotal	-0.874
## modalityConditionmulti:conditionVisual:matcherRespondsTRUE	0.666
## modalityConditionmulti:conditionVisual:incorrectTRUE	0.941
## modalityConditionvocal:conditionVisual:incorrectTRUE	1.503


```
##
## Correlation matrix not shown by default, as p = 32 > 12.
## Use print(x, correlation=TRUE) or
##      vcov(x)          if you need it

## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

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).

```
MuMIn::r.squaredGLMM(finalModel)
```

```
## Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help
## page.
```

```
##           R2m           R2c
## [1,] 0.3466664 0.6931262
```

Fixed effects explain 34.67% of the variance. Total variance explained = 69.31%. (random effects = 34.65%).

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

```
mList = list("m0"=m0,"mtchTrn"=mtchTrn,"modality"=modality,"cond"=cond,
"game"=game,"modXcond"=modXcond,"conXgame"=conXgame,"modXgame"=modXgame,
"moXcoXga"=moXcoXga,"tMtchTr"=tMtchTr,"tMaTxMod"=tMaTxMod,"nTurnXmo"=nTurnXmo,
"nTurnXco"=nTurnXco,"tuXmoXco"=tuXmoXco,"incor"=incor,
"moXincor"=moXincor,"coXincor"=coXincor,"coXmoXin"=coXmoXin,
"multim"=multim,"multiXco"=multiXco,"gamQuad"=gamQuad,
"modXgamQ"=modXgamQ,"block"=block,"blocXmod"=blocXmod)
mList = mList[rownames(modelComparison)]
varExplained = sapply(mList,MuMIn::r.squaredGLMM)
varExplained.fixed = diff(varExplained[1,])
t(t(varExplained.fixed[c("game","gamQuad","mtchTrn",
"tMtchTr","incor","modXcond",
"modXgame","modXgamQ")]))
```

```
##           [,1]
## game      0.108670034
## gamQuad   0.020150604
## mtchTrn   0.112321829
## tMtchTr  -0.001109019
## incor     0.038508550
## modXcond  0.039721398
## modXgame  0.003964707
## modXgamQ  0.001637667
```

Plot for cumulative matcher turn effects (ignore vocal condition who hardly respond). Note that we found that the cumulative number of trials is a significant predictor, except when the number of trials is included in the model. This may be because the total number of trials is an upper bound on the total number of matcher responses and so the model uses cumulative matcher turns as a proxy for number of trials. In the raw data, there appears to be no relationship between cumulative matcher responses and trial length:

```
ggplot(d[d$modalityCondition!="vocal",],  
  aes(x=matcherResponds.cumulative,  
    y=trialLength.log)) +  
  stat_smooth() + geom_point()
```

Alternative models

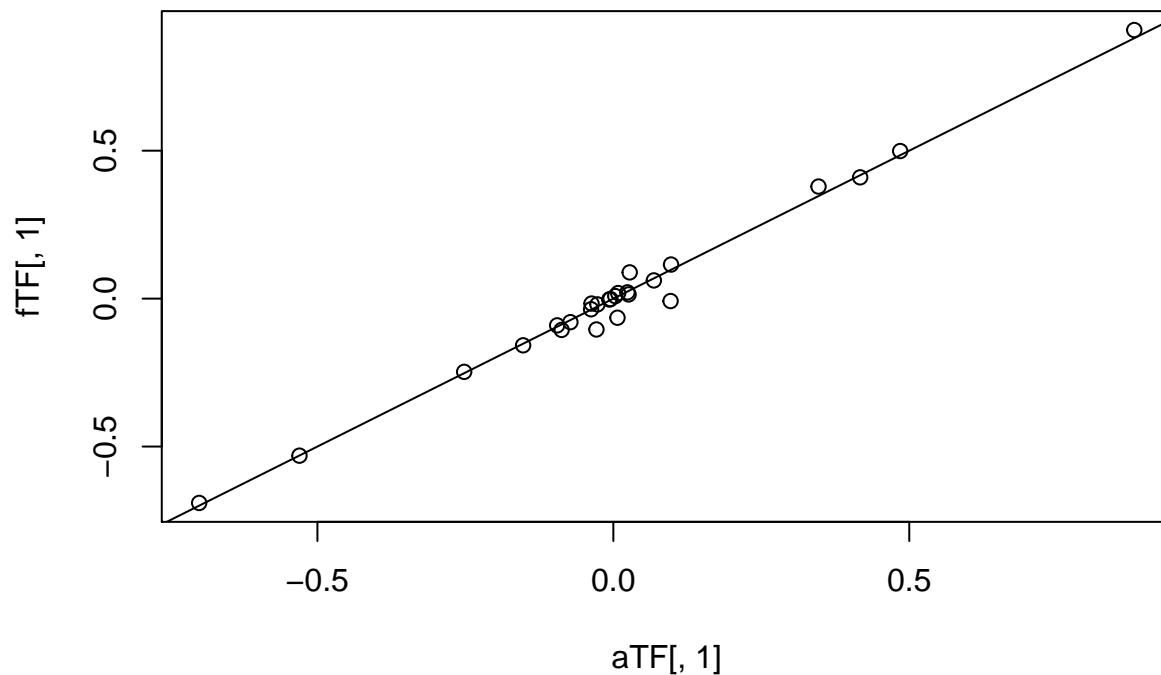
Model for only accurate trials

```
accTrials = lmer(trialLength.log ~ 1 + modalityCondition*condition*trialTotal +  
  I(trialTotal^2) +(modalityCondition:I(trialTotal^2)) +  
  matcherResponds*modalityCondition*condition +  
  matcherResponds.cumulative +  
  matcherResponds.cumulative:modalityCondition +  
  modalityCondition*condition +  
  multimodal + multimodal:condition +  
  matcherResponds +  
  firstBlock +  
  (1 + condition | dyadNumber/playerId) +  
  (1 + modalityCondition|itemId),  
  data=d[!d$incorrect,], REML = TRUE)
```

```
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

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

```
aTF = summary(accTrials)$coefficients  
fTF = summary(finalModel)$coefficients  
cF = intersect(rownames(aTF), rownames(fTF))  
aTF = aTF[cF,]  
fTF = fTF[cF,]  
plot(aTF[,1], fTF[,1])  
abline(0,1)
```



```
# Print large differences in t-value  
diffF = abs(aTF[, 't value'] - fTF[, 't value'])  
diffFV = names(diffF[diffF > (2 * sd(fixef(finalModel))))]  
round(cbind(all=fTF[diffFV,3], correct=aTF[diffFV,3]), 3)
```

##	all correct	
## matcherRespondsTRUE	9.841	9.172
## modalityConditionvisual:matcherRespondsTRUE	-0.068	0.739
## conditionVisual:multimodalTRUE	-0.602	0.060

There are no large qualitative differences when analysing only correct trials.

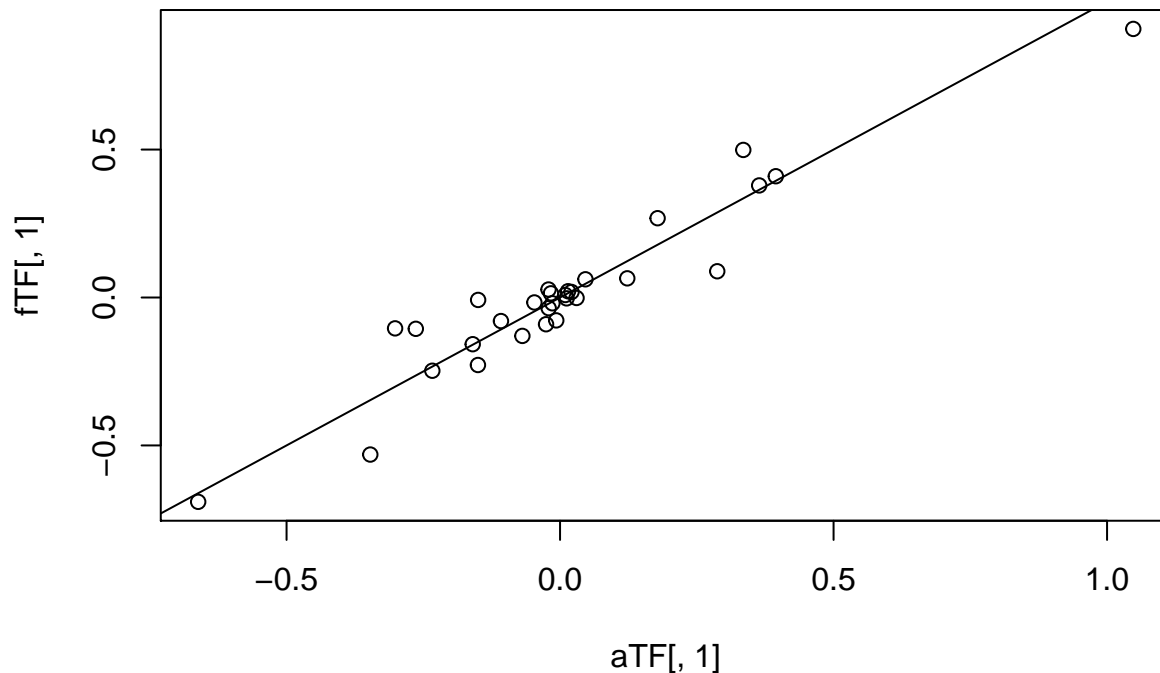
Model for only multimodal trials

```
multTrials = lmer(trialLength.log ~ 1 + modalityCondition*condition*trialTotal +
  I(trialTotal^2) +(modalityCondition:I(trialTotal^2)) +
  matcherResponds*modalityCondition*condition +
  matcherResponds.cumulative +
  matcherResponds.cumulative:modalityCondition +
  incorrect *modalityCondition*condition +
  matcherResponds +
  firstBlock +
  (1 + condition + incorrect |dyadNumber/playerId) +
  (1 + modalityCondition|itemId),
  data=d[(d$modalityCondition!="multi") | (d$multimodal),], REML = TRUE)
```

```
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

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

```
aTF = summary(multTrials)$coefficients
fTF = summary(finalModel)$coefficients
cF = intersect(rownames(aTF),rownames(fTF))
aTF = aTF[cF,]
fTF = fTF[cF,]
plot(aTF[,1],fTF[,1])
abline(0,1)
```



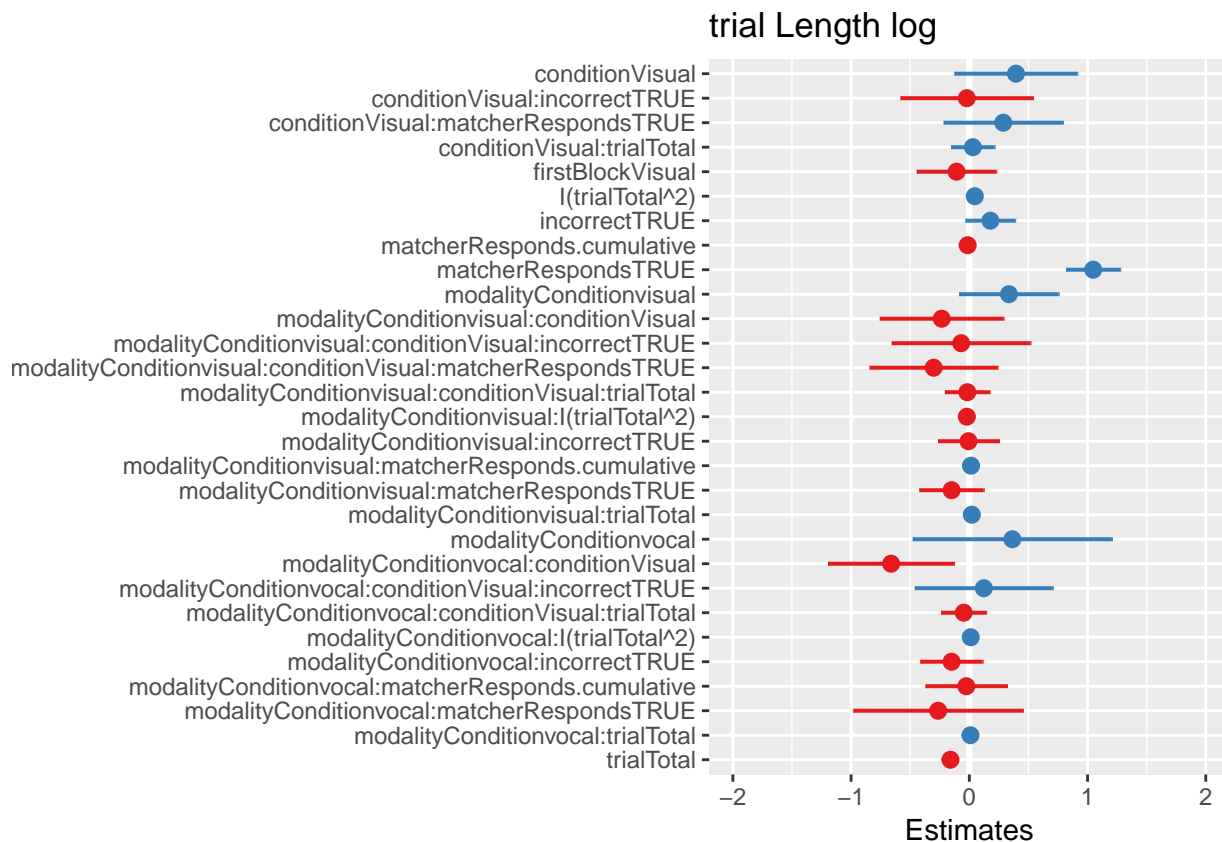
```
# Print large differences in t-value
diffF = abs(aTF[, 't value'] - fTF[, 't value'])
diffFV = names(diffF[diffF > (2 * sd(fixef(finalModel)))]))
round(cbind(all=fTF[diffFV,3],correct=aTF[diffFV,3]),3)
```

##	all	correct
## (Intercept)	-3.055	-1.797
## modalityConditionvisual	2.530	1.560

```
## conditionVisual 3.097 1.487
## trialTotal -8.107 -6.193
## I(trialTotal^2) 4.859 2.370
## matcherRespondsTRUE 9.841 8.993
## matcherResponds.cumulative -1.555 -0.475
## incorrectTRUE 2.926 1.666
## modalityConditionvisual:conditionVisual -1.730 -0.874
## modalityConditionvocal:conditionVisual -4.393 -2.434
## modalityConditionvisual:I(trialTotal^2) -2.063 -0.919
## modalityConditionvocal:I(trialTotal^2) -0.168 0.530
## modalityConditionvisual:matcherRespondsTRUE -0.068 -1.082
## modalityConditionvisual:matcherResponds.cumulative 1.318 0.474
## modalityConditionvisual:incorrectTRUE -0.638 -0.053
## modalityConditionvocal:incorrectTRUE -1.825 -1.118
## modalityConditionvisual:conditionVisual:trialTotal 0.392 -0.173
## modalityConditionvisual:conditionVisual:incorrectTRUE -0.941 -0.230
```

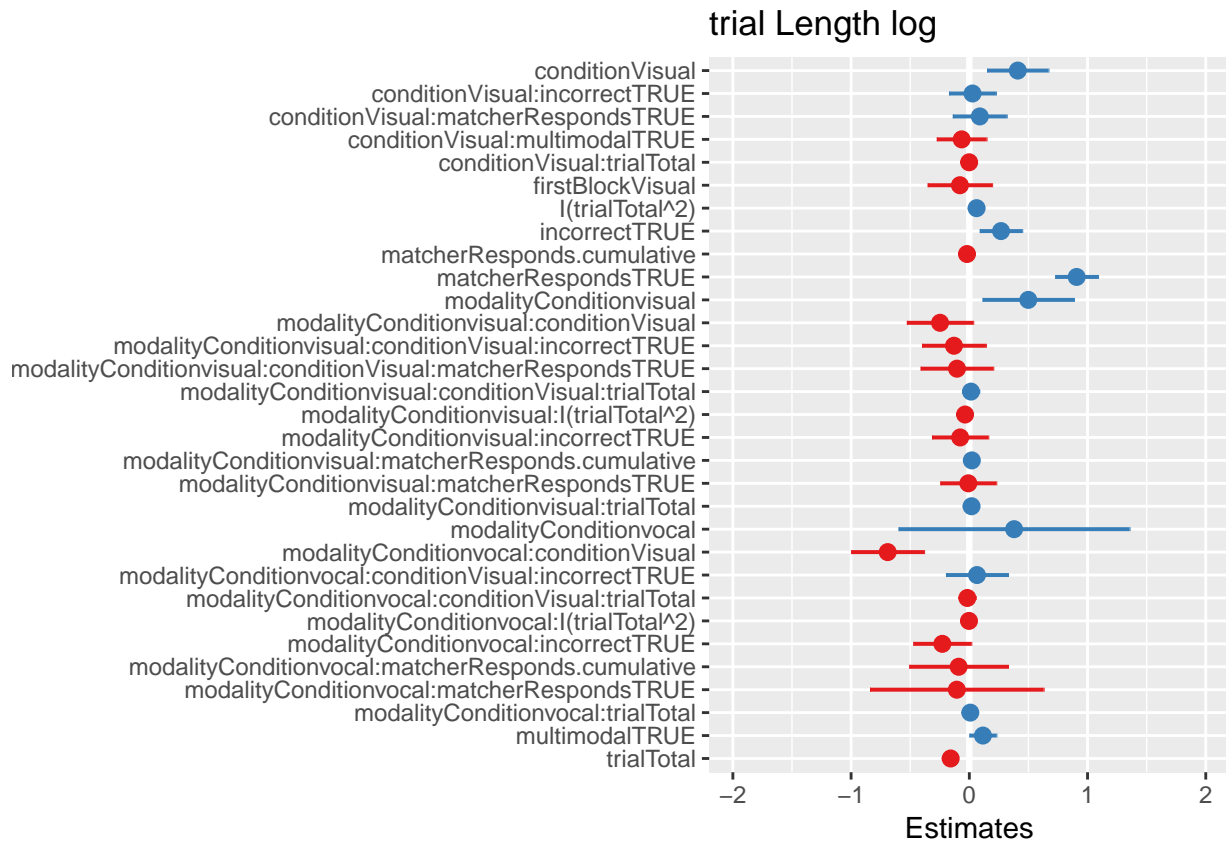
```
plot_model(multTrials,axis.labels = "")
```

```
## Computing p-values via Wald-statistics approximation (treating t as Wald z).
```



```
plot_model(finalModel,axis.labels = "")
```

```
##Computing p-values via Wald-statistics approximation (treating t as Wald z).
```



Summary:

We also analysed only correctly guessed trials and found no qualitative differences. Running the analysis while excluding trials with unimodal descriptions from the multimodal condition lead to weaker effects for condition, stimulus type and the interaction between the two. In line with the analysis above, this suggests that the advantage for participants in the multimodal condition is not just due to multimodal signals in themselves, but may be due to the relationship between multimodality and the communicative interaction between the director and matcher (see section 6.1).

Minimal model

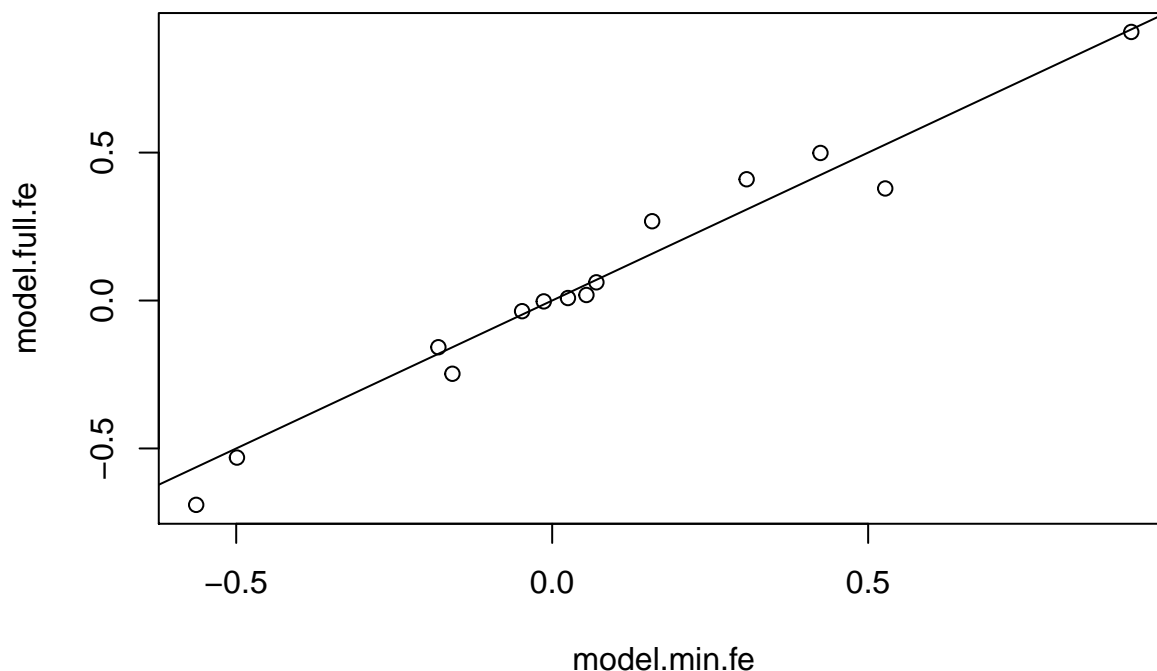
Model including only variables that explained a significant amount of variance according to the model comparison procedure above.

```
model.min = lmer(trialLength.log ~ 1 +  
  matcherResponds +  
  incorrect +  
  modalityCondition:trialTotal +  
  trialTotal + I(trialTotal^2) +  
  modalityCondition*condition +  
  modalityCondition:I(trialTotal^2) +  
  (1 + condition | dyadNumber/playerId) +  
  (1 + modalityCondition|itemId),  
  data=d, REML = FALSE, control = ctrl)
```

boundary (singular) fit: see ?isSingular

Compare the estimates:

```
model.min.fe = fixef(model.min)  
model.full.fe = fixef(finalModel)[names(model.min.fe)]  
plot(model.min.fe,model.full.fe)  
abline(0,1)
```



```
cor(model.min.fe,model.full.fe)
```

```
## [1] 0.9839989
```

Estimates are correlated with $r = 0.98$, suggesting that there is little difference.

Compare the t-values. Note that in almost all cases, the results for the minimal model are the same or stronger:

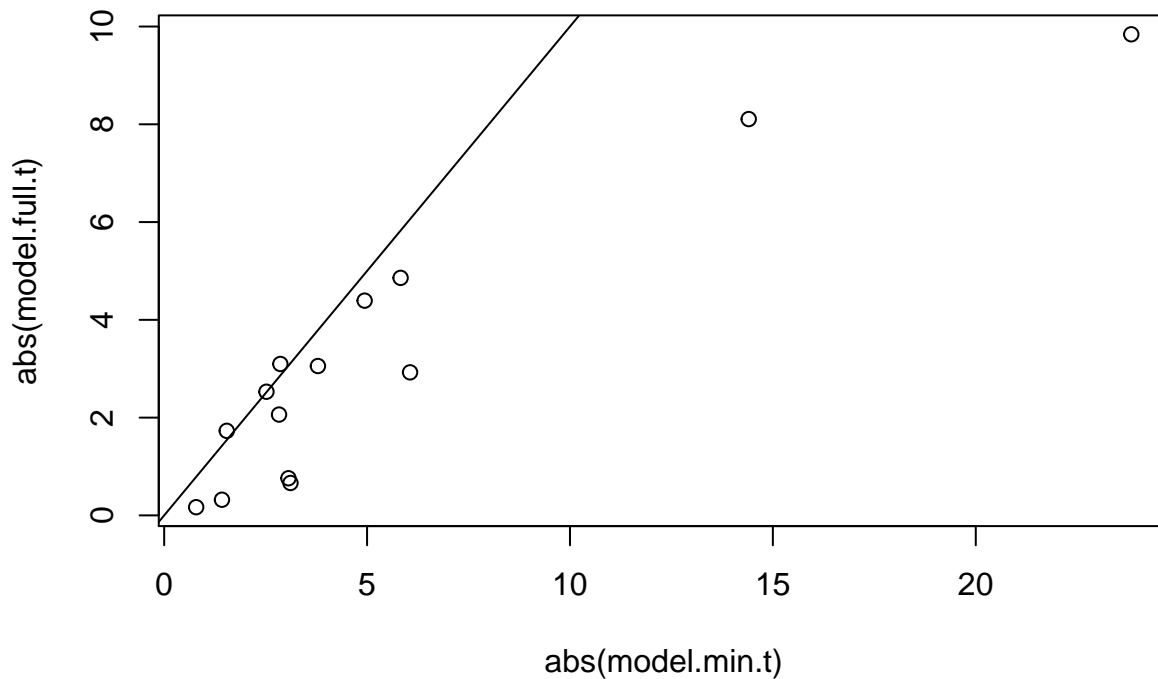
```
getTval = function(X){summary(X)$coef[, "t value"]}  
model.min.t = getTval(model.min)
```



```

model.full.t = getTval(finalModel)
model.full.t = model.full.t[names(model.min.t)]
plot(abs(model.min.t),abs(model.full.t))
abline(0,1)

```



Alternative distribution

Below we run the final model from the procedure above, but fitting the raw trial length in milliseconds with a poisson distribution, rather than the log-transformed trial Length.

```

block.poisson = glmer(trialLength ~ 1 +
  modalityCondition*condition*trialTotal +
  I(trialTotal^2) +(modalityCondition:I(trialTotal^2)) +
  matcherResponds*modalityCondition*condition +
  matcherResponds.cumulative +
  matcherResponds.cumulative:modalityCondition +
  incorrect * modalityCondition*condition +
  multimodal + multimodal:condition +
  matcherResponds +
  firstBlock +
  (1 + condition + incorrect |dyadNumber/playerId) +
  (1 + modalityCondition|itemId),
  data=d, REML = TRUE,
  family = poisson())

```

```
## Warning: extra argument(s) 'REML' disregarded
```

```
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

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

```
summary(block.poisson)
```

```

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: poisson ( log )
## Formula: trialLength ~ 1 + modalityCondition * condition * trialTotal +
## I(trialTotal^2) + (modalityCondition:I(trialTotal^2)) + matcherResponds *
## modalityCondition * condition + matcherResponds.cumulative +
## matcherResponds.cumulative:modalityCondition + incorrect *
## modalityCondition * condition + multimodal + multimodal:condition +
## matcherResponds + firstBlock + (1 + condition + incorrect |
## dyadNumber/playerId) + (1 + modalityCondition | itemId)
## Data: d
##
##      AIC      BIC   logLik deviance df.resid
## 2386781 2387058 -1193340 2386681      1832
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -177.99  -21.51   -5.35   13.47  390.03
##
## Random effects:
##      Groups                Name                Variance Std.Dev.  Corr
## playerId:dyadNumber (Intercept)                0.04897  0.2213
##                      conditionVisual            0.04350  0.2086  -0.63
##                      incorrectTRUE              0.04639  0.2154  -0.22 -0.09
## itemId              (Intercept)                0.03748  0.1936
##                      modalityConditionvisual    0.01607  0.1268  -0.17
##                      modalityConditionvocal    0.01812  0.1346  -0.44  0.62
## dyadNumber          (Intercept)                0.05725  0.2393
##                      conditionVisual            0.01012  0.1006  -0.11
##                      incorrectTRUE              0.01050  0.1025  -0.99  0.21
## Number of obs: 1882, groups:
## playerId:dyadNumber, 30; itemId, 16; dyadNumber, 15
##
## Fixed effects:
##
##                                     Estimate
## (Intercept)                        8.227e+00
## modalityConditionvisual              6.810e-01
## modalityConditionvocal              5.839e-01
## conditionVisual                    5.409e-01
## trialTotal                        -1.760e-01
## I(trialTotal^2)                     8.757e-02
## matcherRespondsTRUE                 8.988e-01
## matcherResponds.cumulative          -2.413e-03
## incorrectTRUE                      3.749e-01
## multimodalTRUE                     2.977e-01
## firstBlockVisual                   -5.543e-02
## modalityConditionvisual:conditionVisual -3.968e-01
## modalityConditionvocal:conditionVisual -7.966e-01
## modalityConditionvisual:trialTotal     3.335e-02
## modalityConditionvocal:trialTotal     3.861e-02
## conditionVisual:trialTotal           -2.647e-02
## modalityConditionvisual:I(trialTotal^2) -6.819e-02
## modalityConditionvocal:I(trialTotal^2) -2.165e-02
## modalityConditionvisual:matcherRespondsTRUE -4.436e-02

```

## modalityConditionvocal:matcherRespondsTRUE	-1.707e-01
## conditionVisual:matcherRespondsTRUE	-2.626e-03
## modalityConditionvisual:matcherResponds.cumulative	-1.506e-05
## modalityConditionvocal:matcherResponds.cumulative	-7.703e-02
## modalityConditionvisual:incorrectTRUE	-1.242e-01
## modalityConditionvocal:incorrectTRUE	-3.383e-01
## conditionVisual:incorrectTRUE	-1.338e-01
## conditionVisual:multimodalTRUE	-2.011e-01
## modalityConditionvisual:conditionVisual:trialTotal	4.511e-02
## modalityConditionvocal:conditionVisual:trialTotal	-3.480e-02
## modalityConditionvisual:conditionVisual:matcherRespondsTRUE	5.790e-02
## modalityConditionvisual:conditionVisual:incorrectTRUE	-1.170e-01
## modalityConditionvocal:conditionVisual:incorrectTRUE	1.999e-01
##	Std. Error
## (Intercept)	1.651e-01
## modalityConditionvisual	1.869e-01
## modalityConditionvocal	4.904e-01
## conditionVisual	1.257e-01
## trialTotal	6.958e-04
## I(trialTotal^2)	4.159e-04
## matcherRespondsTRUE	2.264e-03
## matcherResponds.cumulative	3.877e-04
## incorrectTRUE	8.215e-02
## multimodalTRUE	2.165e-03
## firstBlockVisual	1.298e-01
## modalityConditionvisual:conditionVisual	1.298e-01
## modalityConditionvocal:conditionVisual	1.382e-01
## modalityConditionvisual:trialTotal	9.365e-04
## modalityConditionvocal:trialTotal	8.706e-04
## conditionVisual:trialTotal	8.373e-04
## modalityConditionvisual:I(trialTotal^2)	5.395e-04
## modalityConditionvocal:I(trialTotal^2)	5.590e-04
## modalityConditionvisual:matcherRespondsTRUE	2.868e-03
## modalityConditionvocal:matcherRespondsTRUE	7.748e-03
## conditionVisual:matcherRespondsTRUE	2.804e-03
## modalityConditionvisual:matcherResponds.cumulative	4.806e-04
## modalityConditionvocal:matcherResponds.cumulative	2.096e-01
## modalityConditionvisual:incorrectTRUE	1.162e-01
## modalityConditionvocal:incorrectTRUE	1.162e-01
## conditionVisual:incorrectTRUE	3.207e-03
## conditionVisual:multimodalTRUE	3.641e-03
## modalityConditionvisual:conditionVisual:trialTotal	1.102e-03
## modalityConditionvocal:conditionVisual:trialTotal	1.146e-03
## modalityConditionvisual:conditionVisual:matcherRespondsTRUE	3.637e-03
## modalityConditionvisual:conditionVisual:incorrectTRUE	4.194e-03
## modalityConditionvocal:conditionVisual:incorrectTRUE	4.260e-03
##	z value
## (Intercept)	49.816
## modalityConditionvisual	3.644
## modalityConditionvocal	1.191
## conditionVisual	4.305
## trialTotal	-252.928
## I(trialTotal^2)	210.534
## matcherRespondsTRUE	396.971

```

## matcherResponds.cumulative -6.223
## incorrectTRUE 4.564
## multimodalTRUE 137.488
## firstBlockVisual -0.427
## modalityConditionvisual:conditionVisual -3.057
## modalityConditionvocal:conditionVisual -5.764
## modalityConditionvisual:trialTotal 35.609
## modalityConditionvocal:trialTotal 44.344
## conditionVisual:trialTotal -31.608
## modalityConditionvisual:I(trialTotal^2) -126.410
## modalityConditionvocal:I(trialTotal^2) -38.737
## modalityConditionvisual:matcherRespondsTRUE -15.469
## modalityConditionvocal:matcherRespondsTRUE -22.035
## conditionVisual:matcherRespondsTRUE -0.937
## modalityConditionvisual:matcherResponds.cumulative -0.031
## modalityConditionvocal:matcherResponds.cumulative -0.367
## modalityConditionvisual:incorrectTRUE -1.069
## modalityConditionvocal:incorrectTRUE -2.912
## conditionVisual:incorrectTRUE -41.724
## conditionVisual:multimodalTRUE -55.223
## modalityConditionvisual:conditionVisual:trialTotal 40.922
## modalityConditionvocal:conditionVisual:trialTotal -30.369
## modalityConditionvisual:conditionVisual:matcherRespondsTRUE 15.920
## modalityConditionvisual:conditionVisual:incorrectTRUE -27.903
## modalityConditionvocal:conditionVisual:incorrectTRUE 46.925
## Pr(>|z|)
## (Intercept) < 2e-16 ***
## modalityConditionvisual 0.000268 ***
## modalityConditionvocal 0.233819
## conditionVisual 1.67e-05 ***
## trialTotal < 2e-16 ***
## I(trialTotal^2) < 2e-16 ***
## matcherRespondsTRUE < 2e-16 ***
## matcherResponds.cumulative 4.87e-10 ***
## incorrectTRUE 5.02e-06 ***
## multimodalTRUE < 2e-16 ***
## firstBlockVisual 0.669464
## modalityConditionvisual:conditionVisual 0.002237 **
## modalityConditionvocal:conditionVisual 8.22e-09 ***
## modalityConditionvisual:trialTotal < 2e-16 ***
## modalityConditionvocal:trialTotal < 2e-16 ***
## conditionVisual:trialTotal < 2e-16 ***
## modalityConditionvisual:I(trialTotal^2) < 2e-16 ***
## modalityConditionvocal:I(trialTotal^2) < 2e-16 ***
## modalityConditionvisual:matcherRespondsTRUE < 2e-16 ***
## modalityConditionvocal:matcherRespondsTRUE < 2e-16 ***
## conditionVisual:matcherRespondsTRUE 0.348969
## modalityConditionvisual:matcherResponds.cumulative 0.975003
## modalityConditionvocal:matcherResponds.cumulative 0.713263
## modalityConditionvisual:incorrectTRUE 0.285186
## modalityConditionvocal:incorrectTRUE 0.003590 **
## conditionVisual:incorrectTRUE < 2e-16 ***
## conditionVisual:multimodalTRUE < 2e-16 ***
## modalityConditionvisual:conditionVisual:trialTotal < 2e-16 ***

```

```
## modalityConditionvocal:conditionVisual:trialTotal < 2e-16 ***
## modalityConditionvisual:conditionVisual:matcherRespondsTRUE < 2e-16 ***
## modalityConditionvisual:conditionVisual:incorrectTRUE < 2e-16 ***
## modalityConditionvocal:conditionVisual:incorrectTRUE < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

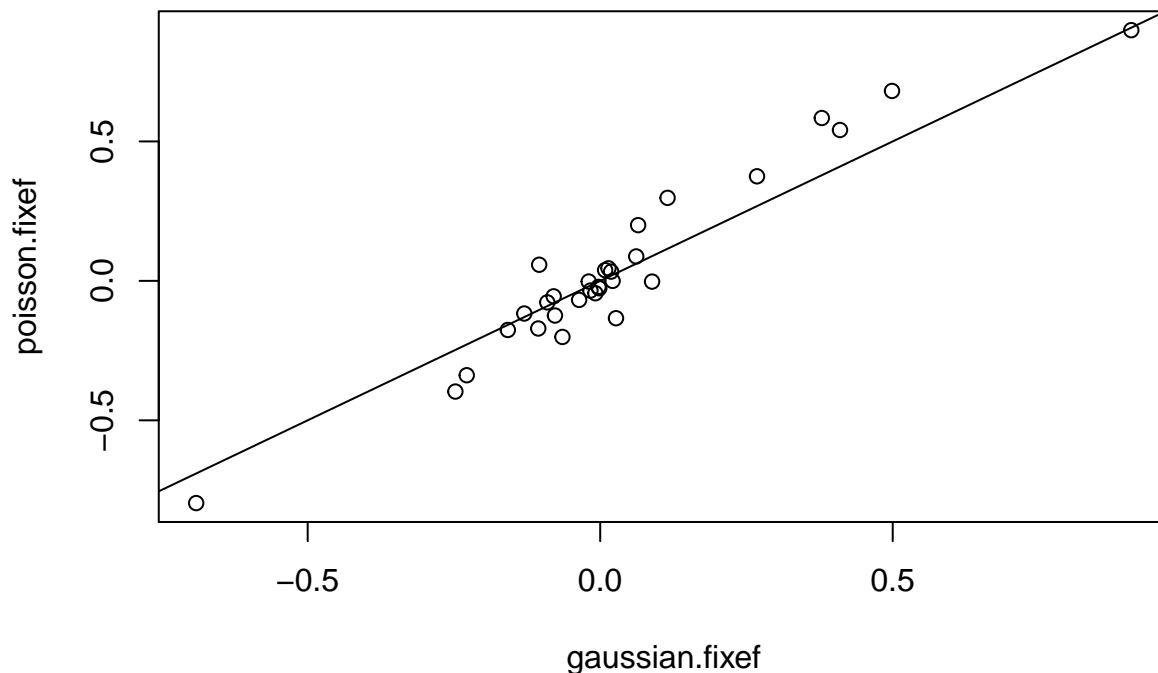
##
## Correlation matrix not shown by default, as p = 32 > 12.
## Use print(x, correlation=TRUE) or
##     vcov(x)         if you need it

## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Compare estimates for the two distribution functions (the intercepts will obviously be different, so we ignore those):

```
gaussian.fixef = fixef(block)
gaussian.fixef = gaussian.fixef[names(gaussian.fixef)!="(Intercept)"]
poisson.fixef = fixef(block.poisson)
poisson.fixef = poisson.fixef[names(poisson.fixef)!="(Intercept)"]

plot(gaussian.fixef,poisson.fixef)
abline(0,1)
```



```
cor(gaussian.fixef,poisson.fixef)
```

```
## [1] 0.9651025
```

The estimates are highly correlated, suggesting that the choice to log-transform the trial length isn't particularly important.

Summary results

```
rownames(finalRes) = NULL
cbind(finalRes[,c("label", "estimate", "conf.low", 'conf.high')],
      signif(finalRes[,c("wald.t", "Chisq", "Pr(>Chisq)"),2])
```

##		label	estimate	conf.low	conf.high
## 1		Visual modality	+4605	847	10136
## 2		Acoustic modality	+3279	-3204	20495
## 3		Visual stimuli	+3607	1157	6784
## 4		Game	-1040	-1268	-803
## 5		Game^2	+452	266	642
## 6		Matcher Responds	+10530	7610	14027
## 7		Total interaction	-138	-309	36
## 8		Incorrect	+2188	659	4019
## 9		Multimodal T1	+869	7	1835
## 10		Visual stims first	-545	-2113	1515
## 11		Visual modality:Visual stimuli	-1561	-2921	239
## 12		Acoustic modality:Visual stimuli	-3551	-4498	-2263
## 13		Visual modality:Game	+134	-255	546
## 14		Acoustic modality:Game	+60	-300	439
## 15		Visual stimuli:Game	-11	-364	360
## 16		Visual modality:Game^2	-252	-483	-13
## 17		Acoustic modality:Game^2	-21	-259	225
## 18		VisualModality:Matcher Responds	-58	-1545	1825
## 19		Vocal Modality:Matcher Responds	-715	-4047	6228
## 20		Visual stim:Matcher Responds	+660	-928	2655
## 21		Total interaction:Visual Modality	+153	-73	385
## 22		Total interaction:Vocal Modality	-615	-2835	2753
## 23		Visual modality:Incorrect	-530	-1924	1238
## 24		Acoustic modality:Incorrect	-1452	-2684	122
## 25		Visual stimuli:incorrect	+195	-1119	1797
## 26		VisualStim:MultimodalT1	-445	-1712	1118
## 27		Visual modality:Visual stimuli:Game	+101	-391	630
## 28		Acoustic modality:Visual stimuli:Game	-118	-594	393
## 29		VisualModality:Visual stim:Matcher Responds	-705	-2401	1600
## 30		Visual modality:Visual stimuli:incorrect	-867	-2347	1074
## 31		Acoustic modality:Visual stimuli:incorrect	+477	-1276	2755
##	wald.t	Chisq	Pr(>Chisq)		
## 1	2.500	0.630	7.3e-01		
## 2	0.760	0.630	7.3e-01		
## 3	3.100	0.540	4.6e-01		
## 4	-8.100	400.000	4.7e-89		
## 5	4.900	79.000	7.2e-19		
## 6	9.800	490.000	7.2e-109		
## 7	-1.600	0.068	7.9e-01		
## 8	2.900	13.000	2.7e-04		
## 9	2.000	1.500	2.2e-01		
## 10	-0.570	0.420	5.2e-01		
## 11	-1.700	15.000	4.4e-04		
## 12	-4.400	15.000	4.4e-04		
## 13	0.660	11.000	5.2e-03		
## 14	0.320	11.000	5.2e-03		

##	15	-0.059	0.190	6.6e-01
##	16	-2.100	6.800	3.3e-02
##	17	-0.170	6.800	3.3e-02
##	18	-0.068	2.900	4.0e-01
##	19	-0.280	2.900	4.0e-01
##	20	0.760	0.140	7.1e-01
##	21	1.300	1.900	4.0e-01
##	22	-0.420	1.900	4.0e-01
##	23	-0.640	5.200	7.5e-02
##	24	-1.800	5.200	7.5e-02
##	25	0.270	0.063	8.0e-01
##	26	-0.600	0.400	5.3e-01
##	27	0.390	0.620	4.3e-01
##	28	-0.470	0.620	4.3e-01
##	29	-0.670	0.860	3.5e-01
##	30	-0.940	2.200	3.4e-01
##	31	0.480	2.200	3.4e-01