Modality effects in a signalling game

Intro

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

Load libraries

```
library(lme4)
```

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

```
##
## Attaching package: 'lme4'
```

```
## The following object is masked from 'package:stats':
##
##     sigma
```

```
library(sjPlot)
```

```
## Visit http://strengejacke.de/sjPlot for package-vignettes.
```

Load data

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

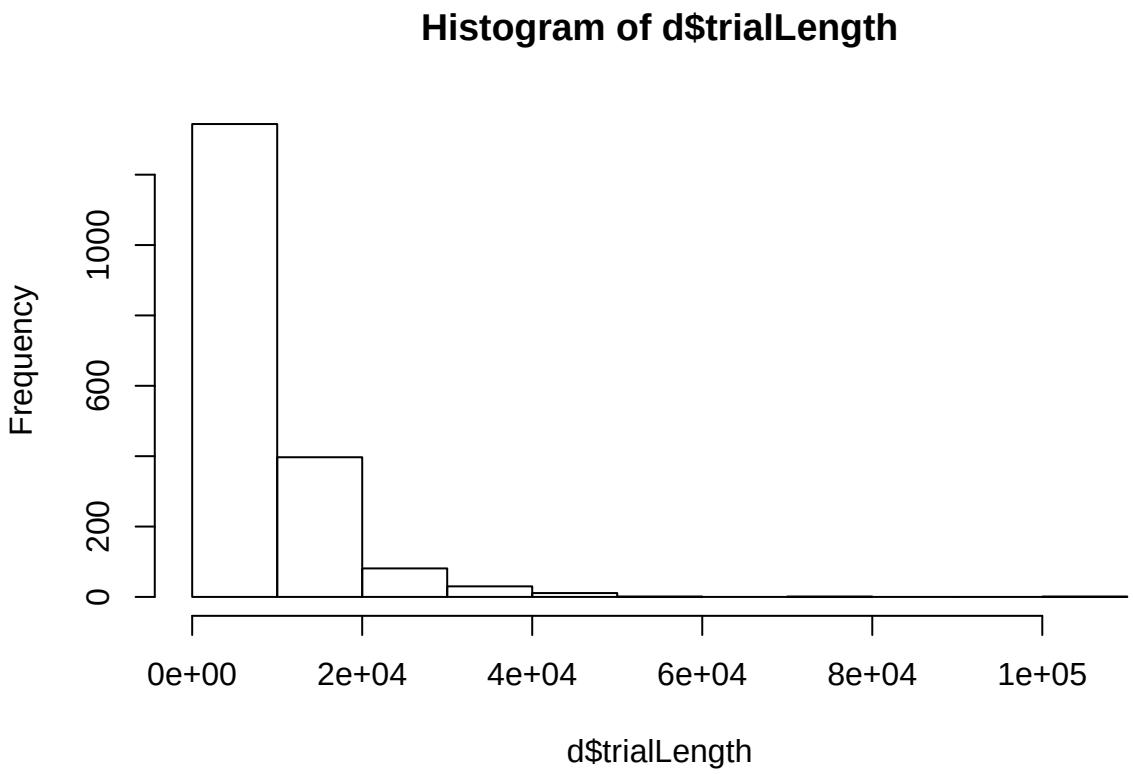
We don't need info on every signal in each turn, just the trial time. Keep only 1st signal in each trial.

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

Descriptive stats

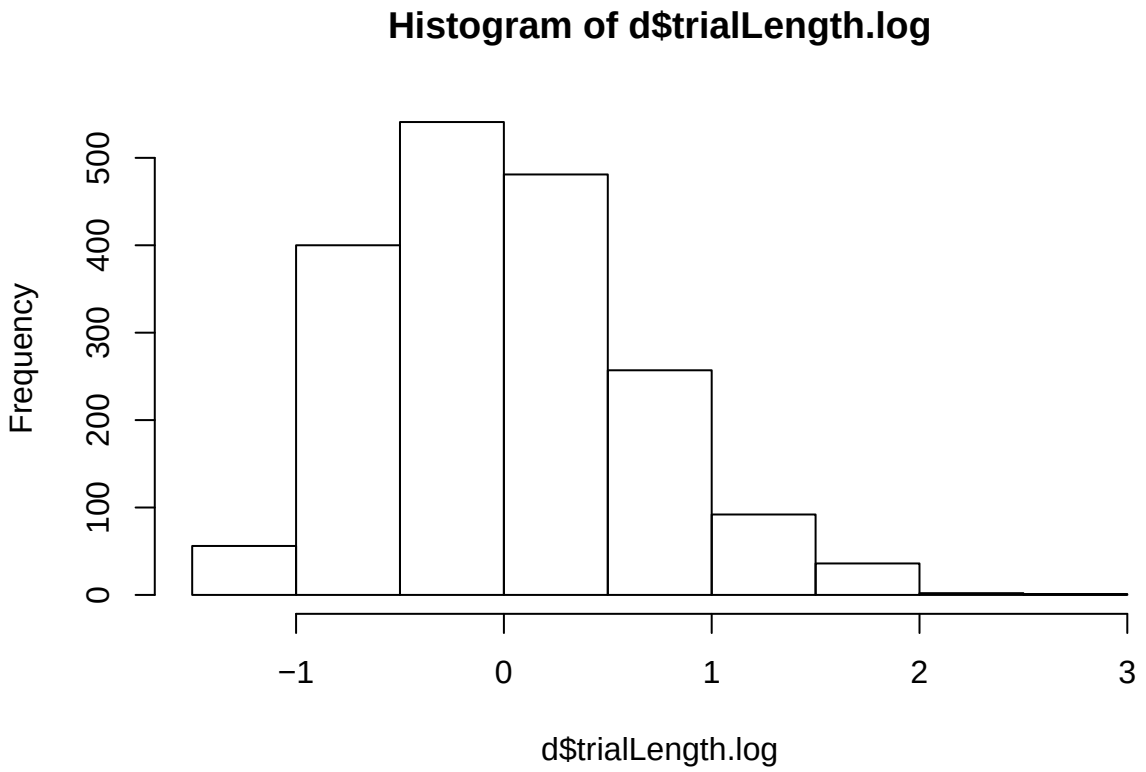
The distribution of trial times is very skewed:

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



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)
```



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

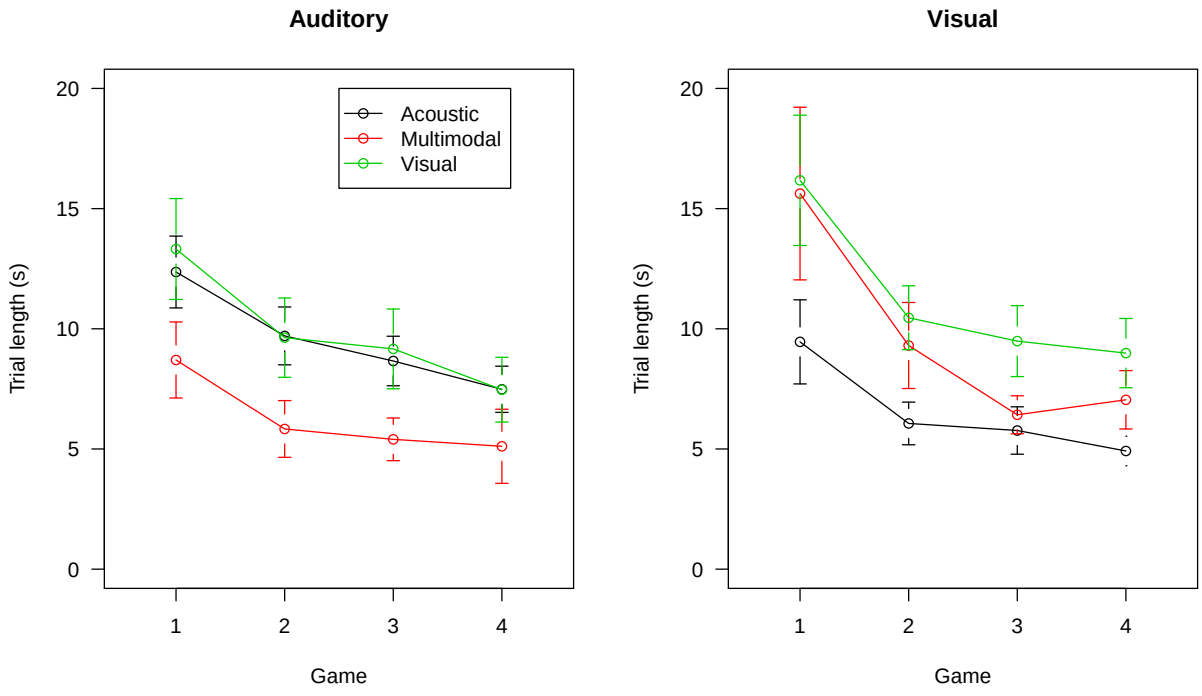


Figure 1: The efficiency of trials in different conditions

Mixed models

Make a series of models with random effects for dyad, director and item.

```
m0 = lmer(trialLength.log ~ 1 +
            (1|dyadNumber) + (1|playerId) + (1|itemId),
          data=d)
m1 = lmer(trialLength.log ~ 1 + modalityCondition +
            (1|dyadNumber) + (1|playerId) + (1|itemId),
          data=d)
m2 = lmer(trialLength.log ~ 1 + modalityCondition + condition +
            (1|dyadNumber) + (1|playerId) + (1|itemId),
          data=d)
m3 = lmer(trialLength.log ~ 1 + modalityCondition + condition + game +
            (1|dyadNumber) + (1|playerId) + (1|itemId),
          data=d)
m4 = lmer(trialLength.log ~ 1 + modalityCondition * condition + game +
            (1|dyadNumber) + (1|playerId) + (1|itemId),
          data=d)
m5 = lmer(trialLength.log ~ 1 + (modalityCondition * condition) + game + (game:condition) +
            (1|dyadNumber) + (1|playerId) + (1|itemId),
          data=d)
m6 = lmer(trialLength.log ~ 1 + modalityCondition * condition * game +
            (1|dyadNumber) + (1|playerId),
          data=d)
```

Results

Compare the fit of the models:

```
anova(m0,m1,m2,m3,m4,m5,m6)
```

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

```
## Data: d
## Models:
## m0: trialLength.log ~ 1 + (1 | dyadNumber) + (1 | playerId) + (1 |
## m0:     itemId)
## m1: trialLength.log ~ 1 + modalityCondition + (1 | dyadNumber) +
## m1:     (1 | playerId) + (1 | itemId)
## m2: trialLength.log ~ 1 + modalityCondition + condition + (1 | dyadNumber) +
## m2:     (1 | playerId) + (1 | itemId)
## m3: trialLength.log ~ 1 + modalityCondition + condition + game +
## m3:     (1 | dyadNumber) + (1 | playerId) + (1 | itemId)
## m4: trialLength.log ~ 1 + modalityCondition * condition + game +
## m4:     (1 | dyadNumber) + (1 | playerId) + (1 | itemId)
## m5: trialLength.log ~ 1 + (modalityCondition * condition) + game +
## m5:     (game:condition) + (1 | dyadNumber) + (1 | playerId) + (1 |
## m5:     itemId)
## m6: trialLength.log ~ 1 + modalityCondition * condition * game +
## m6:     (1 | dyadNumber) + (1 | playerId)
##    Df    AIC    BIC  logLik deviance    Chisq Chi Df Pr(>Chisq)
## m0  5 2933.0 2960.7 -1461.5   2923.0
## m1  7 2934.2 2972.9 -1460.1   2920.2   2.7934      2     0.2474
## m2  8 2935.3 2979.6 -1459.7   2919.3   0.8955      1     0.3440
## m3  9 2614.6 2664.4 -1298.3   2596.6 322.7355      1     <2e-16 ***
## m4 11 2470.1 2530.9 -1224.0   2448.1 148.5021      2     <2e-16 ***
## m5 12 2470.8 2537.2 -1223.4   2446.8   1.3002      1     0.2542
## m6 15 2709.7 2792.7 -1339.8   2679.7   0.0000      3     1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
finalModel = m6
```

Plot the fixed effects

Relabel the effects:

```
feLabels = matrix(c(
"(Intercept)"             ,"Intercept"      ,
"modalityConditionvisual" ,"Visual modality",
"modalityConditionvocal"  , "Acoustic modality",
"conditionVisual" , "Visual stimuli",
"game"             , "Game",
"modalityConditionvisual:conditionVisual"  , "Visual modality:Visual stimuli",
"modalityConditionvocal:conditionVisual" , "Acoustic modality:Visual stimuli",
"modalityConditionvisual:game"    , "Visual modality:Game",
"modalityConditionvocal:game"     , "Acoustic modality:Game",
"conditionVisual:game"             , "Visual stimuli:Game",
"modalityConditionvisual:conditionVisual:game", "Visual modality:Visual stimuli:game",
"modalityConditionvocal:conditionVisual:game", "Acoustic modality:Visual stimuli:game"
```

```
), ncol=2, byrow = T)

feLabels2 = as.vector(feLabels[match(names(fixef(m6)),feLabels[,1]),2])
```

Plot the strength of the fixed effects:

```
sjp.lmer(m6, 'fe',
         show.intercept = T,
         sort.est=NULL,
         axis.labels = feLabels2[2:12],
         xlab="Trial time (ms)",
         geom.colors = c(1))
```

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

## Warning in sj.setGeomColors(me.plot, geom.colors, 2, FALSE, NULL): Too less
## colors provided for plot. Using default color palette.

## Warning: Deprecated, use tibble::rownames_to_column() instead.
```