```
# false alarm rates per condition

setwd('/home/allgoodguys/Documents/Studying/Lund_PhD/epistles/
005_with-Lima/analysis')
library(lme4)
require(brms)
require(shinystan)

df = read.csv ('data/data_preprocessed.csv')[,-1]
# or: df = read.csv('data/data_prepocessed_48sounds.csv')[,-1]

df_distractor = df[df$type=='distractor',] # distractors only
df_distractor$delib = df_distractor$condition == 'Deliberated'

aggregate (falseAlarm~condition, df_distractor, mean)
mod0 = glmer(falseAlarm ~ condition + (1|subject)+(1|item),
family='binomial', data=df_distractor, nAGQ=0)
summary(mod0)
drop1 (mod0, test='Chisq')

mod1 = glmer(falseAlarm ~ delib + (1|subject)+(1|item),
family='binomial', data=df_distractor, nAGQ=0)
summary(mod1)

mod = brm(falseAlarm ~ condition + (1|subject)+(1|item), data =
df_distractor, warmup=100, iter=500, chains=4, cores=4,
family='bernoulli', save_ranef = F, prior=set_prior("normal(0,5)"))
# saveRDS(mod, 'models/mod_falseAlarm~cond.RDS')  or saveRDS(models/
mod, 'mod_falseAlarm~cond_48sounds.RDS')
# mod = readRDS('models/mod_falseAlarm~cond.RDS') or mod =
readRDS('models/mod_falseAlarm~cond_48sounds.RDS')
# stancode(mod)

# plot(mod)
summary(mod)
# launch_shiny(mod)

coda = posterior_samples(mod)
colnames(coda)

a = data.frame(
  delib = coda[,1],
  fast = coda[,1] + coda[,2],
  load1 = coda[,1] + coda[,3],
  load2 = coda[,1] + coda[,4]
)
a = apply (a, 2, function(x)1/(1+exp(-x)))
# write.csv (a, 'output/mcmc_falseAlarm~cond.csv')

df_plot = data.frame (condition=c('Deliberated','Fast','Load
1','Load 2'), lwr=NA, fit=NA, upr=NA)
df_plot[,2:4] = t (apply(a, 2, function(x)quantile(x,probs=c(.
025,.5,.975))*100) )
# write.csv (df_plot, 'output/falseAlarm~cond.csv') or write.csv
```

```
(df_plot, 'output/falseAlarm~cond_48sounds.csv')

# from accuracy to %errors:
df_plot[,2:4] = 100 - df_plot[,2:4]

# contrasts
colnames(a)
quantile (a[,1] - (a[,2]+a[,3]+a[,4])/3, probs=c(.5,.025,.975))*100
# delib vs rest
quantile (a[,1] - a[,3], probs=c(.5,.025,.975))*100 # delib vs load1
quantile (a[,1] - a[,4], probs=c(.5,.025,.975))*100 # delib vs load2
quantile (a[,1] - a[,2], probs=c(.5,.025,.975))*100 # delib vs fast
quantile (a[,3] - a[,4], probs=c(.5,.025,.975))*100 # load 1 vs load
2
quantile (a[,2] - (a[,3]+a[,4])/2, probs=c(.5,.025,.975))*100 # fast
vs load


## emotion
aggregate (falseAlarm~emotionBlock, df_distractor, mean)
mod_em = glmer(falseAlarm ~ emotionBlock + (1|subject)+(1|item),
family='binomial', data=df_distractor, nAGQ=0)
summary(mod_em)
drop1 (mod_em, test='Chisq')
```