```
# hit 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_preprocessed.csv')[,-1]
# or: df = read.csv('data_preprocessed_40sounds.csv')[,-1]

df_target = df[df$type=='target',] # targets only
levels(df$condition)

# df_target$condition = relevel(df_target$condition, ref="Fast") #
change reference level for Wald tests
# df_target$condition = relevel(df_target$condition, ref="Load 1")
aggregate (hit~condition, df_target, mean)
mod0 = glmer(hit ~ condition + (1|subject)+(1|item),
family='binomial', data=df_target, nAGQ=0)
summary(mod0)
drop1 (mod0, test='Chisq')

# delib. vs rest
df_target$delib = df_target$condition == 'Deliberated'
mod_delib = glmer(hit ~ delib + (1|subject)+(1|item),
family='binomial', data=df_target, nAGQ=0)
summary(mod_delib)

mod_pretest = glmer(hit ~ condition + pretestAccuracy + (1|subject)+
(1|item), family='binomial', data=df_target, nAGQ=0)
summary(mod_pretest)
drop1 (mod_pretest, test='Chisq')



mod = brm (hit ~ condition + (1|subject)+(1|item), data = df_target,
warmup=100, iter=500, chains=4, cores=4, family='bernoulli',
ranef=T, prior=set_prior("normal(0,5)"))
# saveRDS(mod, 'models/mod_hit~cond.RDS')  or saveRDS(models/mod,
'mod_hit~cond_40sounds.RDS')
# mod = readRDS('models/mod_hit~cond.RDS') or mod = readRDS('models/
mod_hit~cond_40sounds.RDS')
# stancode(mod)

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

# myfit = fitted (mod,
newdata=data.frame(condition=levels(df_target$condition)),
re_formula=NA, scale='response', robust=T)

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_hit~cond.csv') or write.csv (a,
'output/mcmc_hit~cond_40sounds.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/hit~cond.csv') or write.csv (df_plot,
'output/hit~cond_40sounds.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[,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
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


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





## Does accuracy in pretest influence hit rates and its interaction
with condition in the main experiment?
# pretestAccuracy
mod0 = glmer(hit ~ pretestAccuracy * condition + (1|subject)+(1|
item), family='binomial', data=df_target, nAGQ=0)
summary(mod0)
drop1(mod0, test='Chisq')  # marginal interaction
```

```r
newdata = expand.grid(pretestAccuracy = c(.5, .75, 1),
                      condition = levels(df$condition))
newdata$fit = predict(mod0, newdata, re.form = NA, type =
'response')
newdata$pretestAccuracy = as.factor(newdata$pretestAccuracy)
ggplot(newdata, aes(x = pretestAccuracy, y = fit, group = condition,
fill = condition)) +
  geom_col(position=position_dodge()) +
  ylab('Hit rates in main exp')


mod1 = glmer(hit ~ pretestAccuracy + condition + (1|subject)+(1|
item), family='binomial', data=df_target, nAGQ=0)
summary(mod1)
drop1(mod1, test='Chisq')  # both highly significant

AIC(mod0, mod1)  # mod0 marginally better



## pretestAccuracy_derived
mod2 = glmer(hit ~ pretestAccuracy_derived * condition + (1|subject)
+(1|item), family='binomial', data=df_target, nAGQ=0)
summary(mod2)
drop1(mod2, test='Chisq')  # interaction

newdata = expand.grid(pretestAccuracy_derived = c(.3, .75, 1),
                      condition = levels(df$condition))
newdata$fit = predict(mod2, newdata, re.form = NA, type =
'response')
newdata$pretestAccuracy_derived =
as.factor(newdata$pretestAccuracy_derived)
ggplot(newdata, aes(x = pretestAccuracy_derived, y = fit, group =
condition, fill = condition)) +
  geom_col(position=position_dodge()) +
  ylab('Hit rates in main exp')

mod3 = glmer(hit ~ pretestAccuracy_derived + condition + (1|subject)
+(1|item), family='binomial', data=df_target, nAGQ=0)
summary(mod3)
drop1(mod3, test='Chisq')  # both highly significant

AIC(mod2, mod3)  # mod2 better


### Effect of subjective intensity
mod_subjInt = glmer(hit ~ subjIntensity * condition + (1|subject)
+(1|item), family='binomial', data=df_target, nAGQ=0)
summary(mod_subjInt)
drop1(mod_subjInt, test='Chisq')

# range(df_target$subjIntensity, na.rm = TRUE)
newdata = expand.grid(subjIntensity = c(3, 6),
                      condition = levels(df$condition))
```

```
newdata$fit = predict(mod_subjInt, newdata, re.form = NA, type =
'response')
newdata$subjIntensity = as.factor(newdata$subjIntensity)
ggplot(newdata, aes(x = subjIntensity, y = fit, group = condition,
fill = condition)) +
  geom_col(position=position_dodge()) +
  ylab('Hit rates in main exp')

mod_subjVal = glmer(hit ~ subjValence * condition + (1|subject)+(1|
item), family='binomial', data=df_target, nAGQ=0)
summary(mod_subjVal)
drop1(mod_subjVal, test='Chisq')

mod_subjAr = glmer(hit ~ subjArousal * condition + (1|subject)+(1|
item), family='binomial', data=df_target, nAGQ=0)
summary(mod_subjAr)
drop1(mod_subjAr, test='Chisq')
```