

```

# hit rates per condition and emotion, controlling for pretest
results

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_preprocessed_40sounds.csv')[,-1]

df_target = df[df$type=='target',] # targets only

# with intensity and pretest accuracy as covariates
mod0 = glmer(hit ~ condition*emotionBlock + subjIntensity +
pretestAccuracy + duration + (1|subject)+(1|item),
family='binomial', data=df_target, nAGQ=0)
summary(mod0)
drop1(mod0, test='Chisq')

mod = brm(hit ~ condition*emotionBlock + subjIntensity +
pretestAccuracy + duration + (1|subject)+(1|item), data = df_target,
warmup=300, iter=1000, chains=4, cores=4, family='bernoulli',
save_ranef=F, prior=set_prior("horseshoe(1)", control =
list(adapt_delta = 0.95)) # set_prior for all beta-coef at once
(shrinkage): see https://cran.r-project.org/web/packages/brms/vignettes/brms.pdf
# saveRDS(mod, 'models/mod_hit~cond*em+int+acc+dur_shrinkage.RDS') #
or saveRDS(mod, 'models/
mod_hit~cond*em+int+acc+dur_shrinkage_40sounds.RDS')
# mod = readRDS('models/mod_hit~cond*em+int+acc+dur_shrinkage.RDS')
# or mod = readRDS('models/
mod_hit~cond*em+int+acc+dur_shrinkage_40sounds.RDS')
# stancode(mod)

# plot(mod)
summary(mod)

newdata = expand.grid(
  condition = levels(df_target$condition),
  emotionBlock = levels(df_target$emotionBlock),
  subjIntensity = median(df_target$subjIntensity),
  pretestAccuracy = median(df_target$pretestAccuracy),
  duration = median(df_target$duration)
)
a = fitted(mod, newdata=newdata, summary = FALSE, re_formula = NA)
colnames(a) = paste0(newdata$condition, '_', newdata$emotionBlock)
delib_vs_rest = data.frame(ems = levels(df_target$emotionBlock),
fit=NA, lwr=NA, upr=NA, stringsAsFactors=F)
for (em in 1:nrow(delib_vs_rest)){
  r = 4*em-3
  delib_vs_rest[em,2:4] = quantile(a[,r]-(a[,r+1]+a[,r+2]+a[,r+3])/
3, probs=c(.5,.025,.975)) * 100
}

```

```

}
delib_vs_rest

fast_vs_load12 = data.frame(ems = levels(df_target$emotionBlock),
fit=NA, lwr=NA, upr=NA, stringsAsFactors=F)
for (em in 1:nrow(fast_vs_load12)){
  r = 4*em-3
  fast_vs_load12[em,2:4] = quantile(a[,r+1]-(a[,r+2]+a[,r+3])/2,
probs=c(.5,.025,.975)) * 100
}
fast_vs_load12

load1_vs_load2 = data.frame(ems = levels(df_target$emotionBlock),
fit=NA, lwr=NA, upr=NA, stringsAsFactors=F)
for (em in 1:nrow(load1_vs_load2)){
  r = 4*em-3
  load1_vs_load2[em,2:4] = quantile(a[,r+2]-a[,r+3], probs=c(.
5,.025,.975)) * 100
}
load1_vs_load2

```