Pilot data analysis

valeriia beliaeva

11/8/2021

```r
.libPaths('C:/Users/vbeliaev/Documents/r_packages')

knitr::opts_chunk$set(echo = TRUE)

rm(list = ls())

library(ggsignif)
library(ggplot2)
library(gridExtra)
library(lme4)
library(plyr)
library(dplyr)
library(boot)
library(data.table)

'%!in%' <- function(x,y)!('%in%'(x,y))
```

Data analysis Part I

Behavioral calibration

Data analysis is separated into 2 parts:
1) we test whether the task is working correctly (Behavioral calibration)
2) we test the effect of TI stimulation on choice consistency / accuracy
(vmPFC TI neuromodulation)

In this file the first part of the data analysis is presented.

Main dependent variable is choice consistency / accuracy

Correct answers (variable corr) correspond to trials, in which participant has chosen a picture, which was earlier rated higher during the rating task. For example, participant during the rating task estimated taste of banana for 0.8 and Twix for 0.9. Then in the choice task, when twix and banana are presented in the same trial, if participant chooses Twix, answer is correct, if they choose banana the answer is considered incorrect.

```r
# loading preprocessed data

setwd('C:/Users/vbeliaev/Documents/TI_fMRI/pre_registration_paper/methods/pilot_scripts_paper')
tab1 = as.data.table(read.csv("Data_collected_together_preprocessed.csv"))

dataVal = tab1[Cue_Taste1_Size2 == 1]
dataVal$Taste_diff.abs = scale(abs(dataVal$Taste_diff))
dataVal$Size_diff.abs = scale(abs(dataVal$Size_diff))
```

```
dataPer = tab1[Cue_Taste1_Size2 == 2]
dataPer$Taste_diff.abs = scale(abs(dataPer$Taste_diff))
dataPer$Size_diff.abs = scale(abs(dataPer$Size_diff))

ns = length(unique(tab1$Participant))
```

RUN MODELS

First, verifying impact of taste difference on accuracy in value-based trials.

```
m1.val = glmer(corr ~ Size_diff.abs + Taste_diff.abs + (1+Size_diff.abs + Taste_diff.abs|Participant), d
s1 = summary(m1.val)

s1
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: corr ~ Size_diff.abs + Taste_diff.abs + (1 + Size_diff.abs +
##     Taste_diff.abs | Participant)
##    Data: dataVal
##
##      AIC      BIC   logLik deviance df.resid
##   4413.8   4470.5  -2197.9   4395.8     4035
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.1363 -1.0542  0.4375  0.6762  1.0348
##
## Random effects:
##  Groups      Name           Variance Std.Dev. Corr
##  Participant (Intercept)    0.193894 0.44033
##              Size_diff.abs  0.008779 0.09369  0.46
##              Taste_diff.abs 0.096953 0.31137  1.00 0.39
## Number of obs: 4044, groups:  Participant, 23
##
## Fixed effects:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.13534    0.10070  11.275   <2e-16 ***
## Size_diff.abs   0.01082    0.04311   0.251    0.802
## Taste_diff.abs  0.64150    0.07690   8.342   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Sz_df.
## Size_dff.bs 0.197
## Tast_dff.bs 0.845  0.152
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Second, verifying impact of size difference on accuracy in perceptual trials.

```
m1.per = glmer(corr ~ Size_diff.abs + Taste_diff.abs + (1+Size_diff.abs + Taste_diff.abs|Participant), d
s2 = summary(m1.per)
```

```r
s2
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: corr ~ Size_diff.abs + Taste_diff.abs + (1 + Size_diff.abs +
##     Taste_diff.abs | Participant)
##    Data: dataPer
##
##      AIC      BIC   logLik deviance df.resid
##   4339.7   4396.4  -2160.9   4321.7     4027
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.8702 -0.9880  0.4174  0.6577  1.1233
##
## Random effects:
##  Groups      Name           Variance Std.Dev. Corr
##  Participant (Intercept)    0.186327 0.43166
##              Size_diff.abs  0.063195 0.25139  0.99
##              Taste_diff.abs 0.006228 0.07892  0.41 0.29
## Number of obs: 4036, groups:  Participant, 23
##
## Fixed effects:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.160444   0.099661  11.644   <2e-16 ***
## Size_diff.abs  0.743511   0.067234  11.059   <2e-16 ***
## Taste_diff.abs 0.001088   0.045445   0.024    0.981
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Sz_df.
## Size_dff.bs 0.800
## Tast_dff.bs 0.170  0.084
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

PLOTS

Plot 1 Now, plot the resulting beta estimates of the models above.

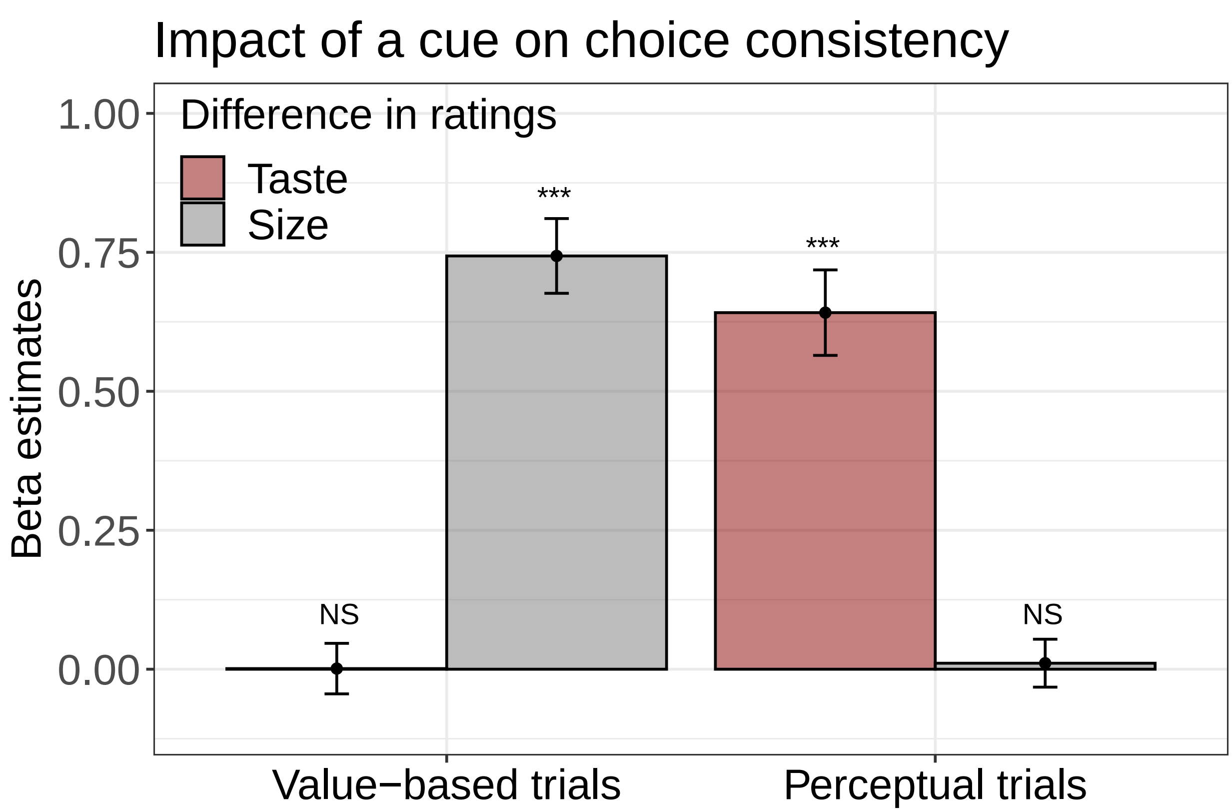
```r
plot_betas = as.data.table(matrix(c(s1$coefficients[2,1], s1$coefficients[3,1],
                      s2$coefficients[2,1], s2$coefficients[3,1],
                      s1$coefficients[2,2], s1$coefficients[3,2],
                      s2$coefficients[2,2], s2$coefficients[3,2]), nrow = 4, ncol = 2))

colnames(plot_betas) = c('beta_mean', 'beta_std')
plot_betas$cue = rep(c("Perceptual trials", "Value-based trials"), each = 2)
plot_betas$cue = factor(plot_betas$cue, levels = c('Value-based trials', 'Perceptual trials'))

plot_betas$ratings = rep(c("Size", "Taste"), 2)
plot_betas$ratings = factor(plot_betas$ratings, levels = c('Taste', 'Size'))
```

```r
p_betas = ggplot(plot_betas, aes(x=cue, y=beta_mean, fill = ratings)) +
  geom_bar(stat="identity", position=position_dodge(), alpha = 0.5, color = 'black') +
  geom_errorbar(aes(ymin=beta_mean-beta_std, ymax=beta_mean+beta_std), width=.1,
              position=position_dodge(.9)) +
  geom_point(position=position_dodge(.9), show_guide = FALSE) +
  theme_bw() +
  ylab('Beta estimates') +
  xlab(' ') +
  scale_fill_manual(values=c('darkred', 'grey48')) +
  ggtitle('Impact of a cue on choice consistency') +
  coord_cartesian(ylim= c(-0.1,1)) +
  theme(text = element_text(size=16), axis.text.x = element_text(size=16, colour = 'black'),
        axis.text.y = element_text(size=16), legend.text=element_text(size=16)) +
  theme(legend.position = c(0.2, 0.87))  +
  guides(fill=guide_legend(title="Difference in ratings")) +
  theme(legend.background=element_blank()) +
  annotate('text', x = 0.78, y = 0.10, label = 'NS') +
  annotate('text', x = 1.22, y = 0.85, label = '***') +
  annotate('text', x = 1.77, y = 0.76, label = '***') +
  annotate('text', x = 2.22, y = 0.10, label = 'NS')

p_betas
```




```r
plot_dir = 'C:/Users/vbeliaev/Documents/TI_fMRI/pre_registration_paper/methods/paper_scripts_v4'
#ggsave('p_betas.png',  path = plot_dir, dpi=300)
```

Plot 2 / probability of choosing top image

Here we plot the dependecy between difference in ratings between the top and the bottom picture and the choice.
In this plot dependent varible is choice top picture (1) or bottom (0).

Differences in ratings of taste or size are grouped in tiles from 1 to 4. Negative tiles represent conditions, when food items presented in the bottom of the screen during the choice task were higher estimated than items shown at the top. The larger is the number of the tile the lager is the difference in ratings between two food items.

```
# Taste trials
tmp_val = ddply(dataVal, .(Participant,Taste_diff.ntile), summarise, acc = mean(Choice01))
tmp_val = ddply(tmp_val, .(Taste_diff.ntile), summarise, acc2 = mean(acc), se=sd(acc)/sqrt(ns))
tmp_val$cue = "Taste"
colnames(tmp_val)[1] = "diff"

tmp_per = ddply(dataVal, .(Participant,Size_diff.ntile), summarise, acc = mean(Choice01))
tmp_per = ddply(tmp_per, .(Size_diff.ntile), summarise, acc2 = mean(acc), se=sd(acc)/sqrt(ns))
tmp_per$cue = "Size"
colnames(tmp_per)[1] = "diff"

tmp_val_per = rbind(tmp_val,tmp_per)

colors =c('darkred', 'grey48')
limits = aes(ymax = acc2+se, ymin=acc2-se, colour=cue, group=cue)
tmp_val_per$diff = factor(tmp_val_per$diff, labels = c(-4,-3,-2,-1, 1, 2, 3, 4))
tmp_val_per$cue = factor(tmp_val_per$cue, levels = c('Taste', 'Size'))

p1.val = ggplot(tmp_val_per, aes(diff, acc2, group=cue)) +
  geom_line(aes(colour=cue), size=2, alpha = 0.5) +
  geom_errorbar(limits, width=0.25, size=1) +
  geom_point(aes(shape=cue, colour=cue), size=4) +
  scale_colour_manual(values=colors) +
  theme_bw() +
  ylab("Probability choose top item") + xlab("Difference in ratings (top item - bottom)") +
  theme(axis.title=element_text(size=10)) + theme(axis.text=element_text(size=10)) +
  theme(legend.text=element_text(size=17)) +
  coord_cartesian(ylim = c(0, 1)) +
  ggtitle("Value-based trials") +
  theme(legend.title = element_blank()) +
  theme(text = element_text(size=16), axis.text.x = element_text(size=16),
        axis.text.y = element_text(size=16)) +
  theme(legend.position = c(0.2, 0.87)) +
  theme(legend.background=element_blank())

#p1.val

# Size trials

tmp_val = ddply(dataPer, .(Participant,Taste_diff.ntile), summarise, acc = mean(Choice01))
tmp_val = ddply(tmp_val, .(Taste_diff.ntile), summarise, acc2 = mean(acc), se=sd(acc)/sqrt(ns))
tmp_val$cue = "Taste"
colnames(tmp_val)[1] = "diff"
```

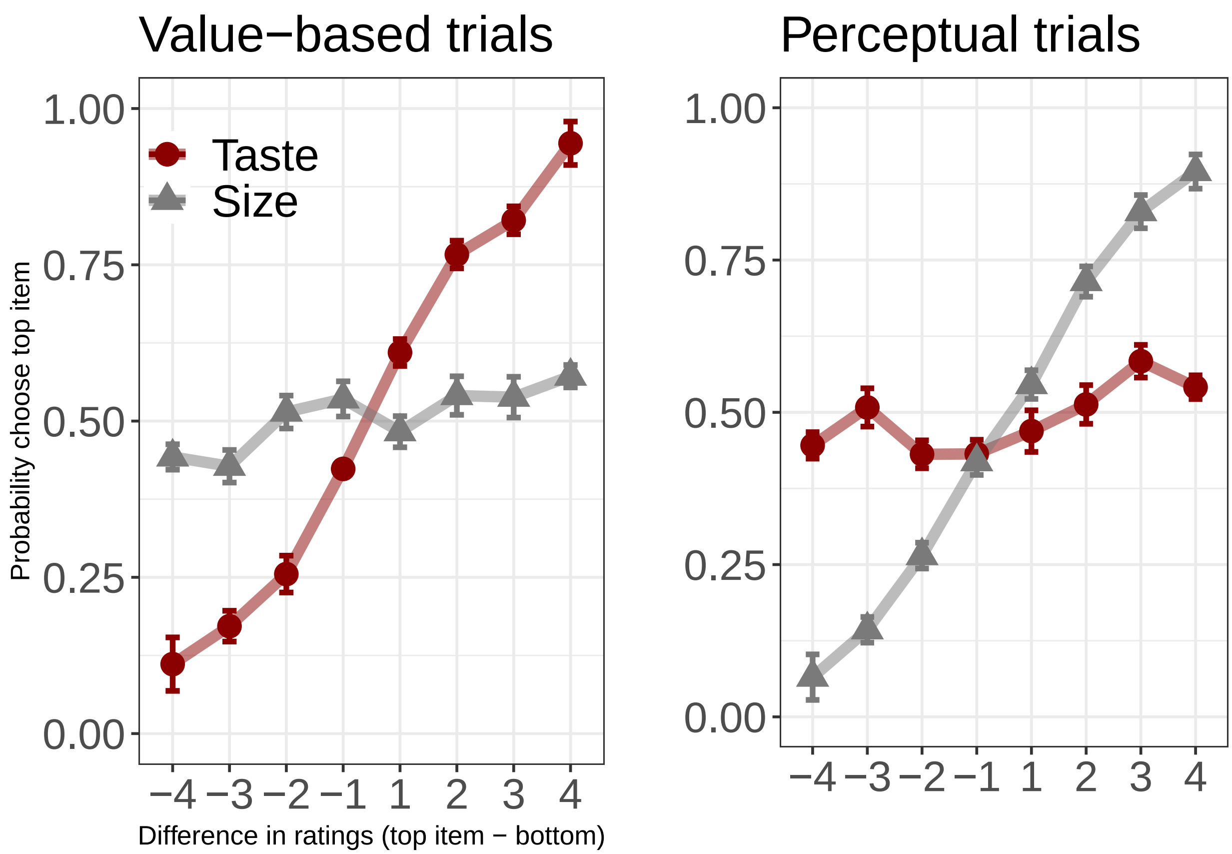
```r
tmp_per = ddply(dataPer, .(Participant,Size_diff.ntile), summarise, acc = mean(Choice01))
tmp_per = ddply(tmp_per, .(Size_diff.ntile), summarise, acc2 = mean(acc), se=sd(acc)/sqrt(ns))
tmp_per$cue = "Size"
colnames(tmp_per)[1] = "diff"

tmp_val_per = rbind(tmp_val,tmp_per)

colors =c('darkred', 'grey48')
limits = aes(ymax = acc2+se, ymin=acc2-se, colour=cue, group=cue)
tmp_val_per$diff = factor(tmp_val_per$diff, labels = c(-4,-3,-2,-1, 1, 2, 3, 4))
tmp_val_per$cue = factor(tmp_val_per$cue, levels = c('Taste', 'Size'))

p1.per = ggplot(tmp_val_per, aes(diff, acc2, group=cue)) +
  geom_line(aes(colour=cue), size=2, alpha = 0.5) +
  geom_errorbar(limits, width=0.25, size=1) +
  geom_point(aes(shape=cue, colour=cue), size=4) +
  scale_colour_manual(values=colors) +
  theme_bw() +
  ylab(" ") +
  xlab(" ") +
  theme(axis.title=element_text(size=17)) + theme(axis.text=element_text(size=17)) +
  theme(legend.text=element_text(size=17)) +
  coord_cartesian(ylim = c(0, 1)) +
  ggtitle("Perceptual trials") +
  theme(legend.title = element_blank()) +
  theme(text = element_text(size=16), axis.text.x = element_text(size=16),
        axis.text.y = element_text(size=16)) +
  theme(legend.position = "none") +
  theme(legend.background=element_blank())

grid.arrange(p1.val, p1.per, ncol=2, widths=c(1,1))
```




```
#g = arrangeGrob(p1.val, p1.per, ncol=2)
#ggsave('p_prob_choose.png',  g, path = plot_dir, dpi=300)
```