

MainAnalysisReport.R

tomra

2022-01-25

```
# Main analysis of CAT experiments with affective faces and depression.  
# Last edit: Tom Salomon, January 2022
```

```
# Load packages  
library(reshape)  
library(ggplot2)  
library(lme4)
```

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

```
##  
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:reshape':  
##  
##      expand
```

```
library(tidyr)
```

```
##  
## Attaching package: 'tidyr'
```

```
## The following objects are masked from 'package:Matrix':  
##  
##      expand, pack, unpack
```

```
## The following objects are masked from 'package:reshape':  
##  
##      expand, smiths
```

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:reshape':  
##  
##      rename
```

```
## The following objects are masked from 'package:stats':
##
##   filter, lag
```

```
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(ggpubr)
```

```
## Registered S3 methods overwritten by 'car':
##   method                      from
##   influence.merMod             lme4
##   cooks.distance.influence.merMod lme4
##   dfbeta.influence.merMod      lme4
##   dfbetas.influence.merMod     lme4
```

```
library(ggbeeswarm)
```

```
## Warning: package 'ggbeeswarm' was built under R version 4.0.5
```

```
library(devEMF)
```

```
# Clear workspace
```

```
rm(list=ls())
```

```
pwd = dirname(rstudioapi::getActiveDocumentContext())$path
```

```
setwd(pwd)
```

```
figures_path = paste0(pwd, "/Figures/")
```

```
# Additional functions ----
```

```
se = function(x) { out=sqrt((var(x, na.rm = TRUE))/(length(which(!is.na(x))))) }
```

```
expit = function(x){exp(x)/(1+exp(x))}
```

```
save_plot = function(plot, file, width = 7, height = 4){
  emf(file = paste0(file, ".emf"), width = width, height = height)
  print(plot)
  dev.off()
  pdf(file = paste0(file, ".pdf"), width = width, height = height)
  print(plot)
  dev.off()
}
```

```
# Pre-allocation ----
```

```
# Data
```

```
Probe_data = list()
```

```
data_by_sub_both = list()
```

```
descriptive_both = list()
```

```
# models
```

```
glmer_model_both = list() # main model of probe effect
```

```
model_summary_both = list()
```

```
model_PHQ_predict_full = list() # predict PHQ by CAT effect for happy faces
```

```
model_PHQ_predict_main = list() # predict PHQ by Binary ranking for happy faces
```

```
# plots
```

```

PHQ_ditribution_plot = list()
barplot_2_groups = list()
scatterplot_PHQ_prediction = list()

# Go over experiments ----
for (exp_i in c(1:2)){
  #### Load data ----
  if (exp_i == 1){
    ## Sample I
    path= paste0(pwd, "/Experiment_01/Output/")
    subjects=c(113,301,303:304,306:307,309:312,314:316,318:326,328:331,333:346,348:351,353:355,357:359)
    #subjects = c(113,132,301:312,314:326,328:331,333:346,348:359)
    # exclude:
    # 132, 302, 305 - Training: Minimal Ladder
    # 308, 317 - Training: false alarm
    # 356 - Binary ranking
    # (technical)
    # % 313 - was coded as 113
    # % 327 - code crashed at BR
    # % 332 - was coded as 132
    # % 347 - quit in the middle of the experiment

    # Consider exclusion
    # 352 - missing one probe run
    # 334-341 - did less training runs
  }
  if (exp_i == 2){
    ## Sample II - Replication
    #For iMac
    path= paste0(pwd, "/Experiment_02/Output/")
    subjects=c(101:102,104:114,116:118,120:122,124:125,127:139,142:152,155:157,159:166,168:181)
    # exclude:
    # 103 - participates in a previous CAT experiment (and had high false alarm in this experiment)
    # 123, 126, 140, 154, 167 - Training: false alarm
    # 158 - Training: misses
    # 115, 158 - Training: minimal ladder
    # 119, 141, 153 - quit in the middle
  }
  # Load demographic data
  filelist=c()
  for (s in subjects){
    filelist=c(filelist, Sys.glob(paste(path, "*", s, "*personalDetails*.txt", sep=""))[1])
  }
  Demographics=c()
  options(warn = -1) # silence warnings of participants with missing demographic info
  for (f in filelist){
    tmp_data=read.delim(f, header=T, na.strings=c(99,999,999000))
    Demographics=rbind(Demographics, tmp_data)
    options()
  }
  options(warn = 0) # unsilence warnings

  # Load probe data

```

```

filelist=c()
for (s in subjects){
  filelist=c(filelist,Sys.glob(paste(path, "*",s,"*probe_block*.txt",sep="")))
}
Probe_Data=c()
for (f in filelist){
  tmp_data=read.table(f,header=T,na.strings=c(99,999,999000,""))
  Probe_Data=rbind(Probe_Data,tmp_data)
}
Probe_Data$PairType2=Probe_Data$PairType
Probe_Data$PairType2[Probe_Data$PairType2>4]=NA # remove trials with both NoGo
Probe_Data$PairType2=factor(Probe_Data$PairType2,labels=c("Happy_Faces","Neutral_Faces","Sanity_Happy"))

# Load PHQ data
PHQ_data=read.table(Sys.glob(paste(path, "/PHQ_Scores.txt",sep="")),header=T)
PHQ_data$PHQ_group=NA
PHQ_data$PHQ_group[PHQ_data$PHQ<=4] = 1 # Low PHQ
PHQ_data$PHQ_group[PHQ_data$PHQ>=10] = 2 # High PHQ
PHQ_data$PHQ_group2=PHQ_data$PHQ_group
PHQ_data$PHQ_group2[is.na(PHQ_data$PHQ_group)] = 1.5 # Intermediate
PHQ_data$PHQ_group=factor(PHQ_data$PHQ_group,labels=c('Low PHQ','High PHQ'))
PHQ_data$PHQ_group2=factor(PHQ_data$PHQ_group2,labels=c('Low PHQ','Intermediate','High PHQ'))
PHQ_data$subjectID = PHQ_data$SubjectID

# Load preference data
Pref_data =c()
Pref_bias =c()
for (s in subjects){
  tmp_file=Sys.glob(paste(path, "*",s,"*ItemRankingResults_combined*.txt",sep=""))
  tmp_data=read.table(tmp_file,header=T,na.strings=c(999,999000))
  differences = tmp_data$Rank[1:40] - tmp_data$Rank[41:80]
  Happy_Faces_Pref_Bias = mean(differences)/sd(differences)
  subjectID = as.character(tmp_data$Subject[1])
  Pref_data = rbind(Pref_data, tmp_data)
  Pref_bias = rbind(Pref_bias,data.frame(subjectID,Happy_Faces_Pref_Bias))
}

# merge PHQ, preferences and probe
Data = left_join(Probe_Data,PHQ_data,by = "subjectID")

data_by_sub=cast(data=Data,formula = subjectID ~ PairType2, value='Outcome' , mean,na.rm=T)
data_by_sub = merge(data_by_sub,PHQ_data,by.x='subjectID',by.y='SubjectID')
data_by_sub$PHQ_rank = rank(data_by_sub$PHQ)
data_by_sub$Happy_Faces_CAT = data_by_sub$Happy_Faces
data_by_sub = left_join(data_by_sub,Pref_bias, by = "subjectID")

# PHQ models ----
model_PHQ_predict_full[[exp_i]]=lm(scale(PHQ) ~ scale(Happy_Faces_CAT) * scale(Happy_Faces_Pref_Bias)
model_PHQ_predict_main[[exp_i]]=lm(scale(PHQ) ~ scale(Happy_Faces_CAT) + scale(Happy_Faces_Pref_Bias)

# Probe model ----
glmer_model = glmer(Outcome ~ 1 + PairType2 * PHQ_group2 + (1 + PairType2|subjectID),
  data=subset(Data,(PairType2<=2 & (!(PHQ_group2=='Intermediate'))),
  na.action=na.omit,family=binomial)

```

```

X_pred = rbind(c(1,0,0,0), # Happy Faces - Low PHQ
               c(1,0,1,0), # Happy Faces - High PHQ
               c(1,1,0,0), # Neutral Faces - Low PHQ
               c(1,1,1,1), # Neutral Faces - High PHQ
               c(0,0,-1,0), # Interaction - Happy Faces: Low PHQ > High PHQ
               c(0,0,-1,-1), # Interaction - Neutral Faces: Low PHQ > High PHQ
               c(0,0,0,1) # Interaction
)
colnames(X_pred) = colnames(model.matrix(glmer_model))

Pred = data.frame(Effect = c("HappyFaces LowPHQ", "HappyFaces HighPHQ", "NeutralFaces LowPHQ", "NeutralFaces HighPHQ",
                             "HappyFaces: Low PHQ > High PHQ", "NeutralFaces: Low PHQ > High PHQ",
                             "Interaction"),
                  PHQ = factor(c(1,2,1,2,NA,NA,NA), labels = c("Low","High")),
                  Stimuli = c("Happy Faces","Happy Faces","Neutral Faces","Neutral Faces",NA,NA,NA),
                  LogOR = X_pred %>% fixef(glmer_model)) %>%
mutate(SE = diag(X_pred %>% vcov(glmer_model)) %>% t(X_pred))0.5,
       Z = LogOR/SE,
       p = pnorm(abs(Z), lower.tail = FALSE)*2,
       OR = exp(LogOR),
       OR_lower = exp(qnorm(p = 0.025)*SE + LogOR),
       OR_upper = exp(qnorm(p = 0.975)*SE + LogOR),
       prop = expit(LogOR),
       prop_lower = expit(qnorm(p = 0.025)*SE + LogOR),
       prop_upper = expit(qnorm(p = 0.975)*SE + LogOR),
       asteriks = ""
)
Pred$asteriks[Pred$p < 0.05] = "*"
Pred$asteriks[Pred$p < 0.01] = "**"
Pred$asteriks[Pred$p < 0.001] = "***"

# print demographics ----
colnames(Demographics)[4] = "Gender"
cat("\nExperiment ", exp_i,
    "\n =====\n",
    "n = ",nrow(Demographics), " (female = ",sum(Demographics$Gender==1),"), ",
    "\nAge: ",min(Demographics$age, na.rm = T)," - ",max(Demographics$age, na.rm = T),
    ", M = ", round(mean(Demographics$age, na.rm = T),2), " (SD = ",round(sd(Demographics$age, na.rm = T),2),
    sep="")

Demographics_full = Demographics %>%
  mutate(SubjectID = subjectID) %>%
  left_join(PHQ_data, by = "subjectID")

Demographics_full %>%
  group_by(PHQ_group2) %>%
  summarise(n = n(), Female = sum(Gender==1),
            mean_age = mean(age, na.rm=T)) %>%
  print()

Probe_data[[exp_i]] = Data
data_by_sub_both[[exp_i]] = data_by_sub
glmer_model_both[[exp_i]] = glmer_model

```

```

model_summary_both[[exp_i]] = Pred
}

```

```

##
## Experiment 1
## =====
## n = 50 (female = 36),
## Age: 19 - 35, M = 24.22 (SD = 3.47)
## # A tibble: 3 x 4
##   PHQ_group2      n Female mean_age
## * <fct>      <int> <int>    <dbl>
## 1 Low PHQ      22     17     23.7
## 2 Intermediate   3      2      24
## 3 High PHQ     25     17     24.7
##
## Experiment 2
## =====
## n = 70 (female = 47),
## Age: 18 - 34, M = 23.71 (SD = 3.19)
## # A tibble: 3 x 4
##   PHQ_group2      n Female mean_age
## * <fct>      <int> <int>    <dbl>
## 1 Low PHQ      19     12     24.5
## 2 Intermediate  29     18      23
## 3 High PHQ     22     17     23.9

```

```

# Build plots ----
for (exp_i in 1:2) {
  Pred = model_summary_both[[exp_i]]
  data_by_sub = data_by_sub_both[[exp_i]]
  ### PHQ distribution ----
  PHQ_ditribution_plot[[exp_i]] =
    ggplot(data_by_sub,aes(PHQ_group2,PHQ,color=PHQ_group2)) +
    theme_bw() +
    geom_quasirandom(size = 2,dodge.width = 0.5, alpha= 0.6, stroke = .3,show.legend = FALSE) +
    geom_quasirandom(size = 2,dodge.width = 0.5, alpha= 1, stroke = .3,
                      color = "black", shape = 21, show.legend = FALSE) +
    # geom_point(size = 2, alpha= 0.6,position = position_jitter(w = 0.4, h = 0.1, seed=2), stroke = .3,
    # geom_point(size = 2, alpha= 1, position = position_jitter(w = 0.4, h = 0.1, seed=2), stroke = .3,
    #           color = "black", shape = 21) +
    geom_hline(yintercept = 9, linetype=2) +
    geom_hline(yintercept = 5, linetype=2) +
    labs(x = "Experimental group", color = "") +
    ggtitle("") +
    ylim(c(-1,24)) +
    theme(text = element_text(size=8), legend.position = "top")

  # PHQ prediction model ----
  data_by_sub = data_by_sub %>%
    mutate(Happy_Faces_Pref_Bias_scaled =
           (Happy_Faces_Pref_Bias - min(Happy_Faces_Pref_Bias))/
           (max(Happy_Faces_Pref_Bias)-min(Happy_Faces_Pref_Bias)))
  data_by_sub_long = rbind(data_by_sub[,c("subjectID","PHQ","PHQ_group")],

```

```

      data_by_sub[,c("subjectID", "PHQ", "PHQ_group")]) %>%
mutate(Predictor = rep(c("CAT Effect", "Preference Bias"), each = nrow(data_by_sub)),
      #mutate(Predictor = rep(c("CAT Effect", "Preference Bias (scaled)"), each = nrow(data_by_sub)),
      #X = c(data_by_sub$Happy_Faces_CAT, data_by_sub$Happy_Faces_Pref_Bias_scaled))
      X = c(data_by_sub$Happy_Faces_CAT, data_by_sub$Happy_Faces_Pref_Bias))
scatterplot_PHQ_prediction[[exp_i]] =
  data_by_sub_long %>% ggplot(aes(x = X, y = PHQ, color=Predictor))+
  facet_grid(. ~ Predictor, scales = "free_x") +
  labs(x = "Independent variable", title = "") +
  geom_smooth(method='lm', formula = 'y ~ x') +
  geom_point(size = 2, alpha= 0.5) +
  geom_point(size = 2, alpha= 1, shape = 21, color = "black") +
  theme_bw()+
  theme(text = element_text(size=10))

# Probe ----
dodge_size = .7
barplot_probe =
  Pred %>% filter(!is.na(PHQ)) %>%
  ggplot(aes(x = Stimuli, y = prop, fill = PHQ)) +
  theme_bw() +
  ggtitle("") +
  geom_bar(width=dodge_size, position=position_dodge(dodge_size), stat="identity") + # Bar plot
  # scale_fill_brewer(palette="Set2") +
  theme(legend.position="top") + # position legend
  geom_errorbar(position=position_dodge(dodge_size), width=dodge_size/4, aes(ymin=prop_lower, ymax=prop_upper)) +
  scale_y_continuous("Proportion of trials Go items were chosen", limit=c(0,1), breaks=seq(0, 1, 0.1), expand=c(0,0)) +
  geom_hline(yintercept = 0.5, linetype = 2, size = 1) + # chace level 50% reference line
  geom_text(position=position_dodge(dodge_size), aes(y=prop_upper+0.01, label=(asteriks)), size=5) # significance

# Add interaction indices
if (Pred$p[5]<0.05) { # Interaction effect: Happy faces
  h1 = max(Pred$prop_upper[1:2])
  InteractionDF_1 = data.frame(x1 = 1 + c(-dodge_size, -dodge_size, dodge_size, dodge_size)/4, y1 = h1 + 0.1,
                              x2 = 1, y2 = h1+0.1, asterisk = Pred$asteriks[5],
                              PHQ = Pred$PHQ[1])
  barplot_probe = barplot_probe +
    geom_path(data = InteractionDF_1, aes(x = x1, y = y1), size = 0.3) +
    geom_text(data = InteractionDF_1[1,], aes(x= x2, y = y2, label = asterisk), size = 5)
}
if (Pred$p[6]<0.05) { # Interaction effect: Neutral faces
  h2 = max(Pred$prop_upper[3:4])
  InteractionDF_2 = data.frame(x1 = 2 + c(-dodge_size, -dodge_size, dodge_size, dodge_size)/4, y1 = h2 + 0.1,
                              x2 = 2, y2 = h2+0.1, asterisk = Pred$asteriks[6],
                              PHQ = Pred$PHQ[1])
  barplot_probe = barplot_probe +
    geom_path(data = InteractionDF_2, aes(x = x1, y = y1), size = 0.3) +
    geom_text(data = InteractionDF_2[1,], aes(x= x2, y = y2, label = asterisk), size = 5)
}
if (Pred$p[7]<0.05) { # Interaction effect
  h3 = max(Pred$prop_upper[1:4])
  InteractionDF_3 = data.frame(x1 = c(1,1,2,2), y1 = h3 + 0.1 + c(.05, 0.08, 0.08, .05), # interaction
                              x2 = 1.5, y2 = 0.1 + h3 + 0.1, asterisk = Pred$asteriks[7],

```

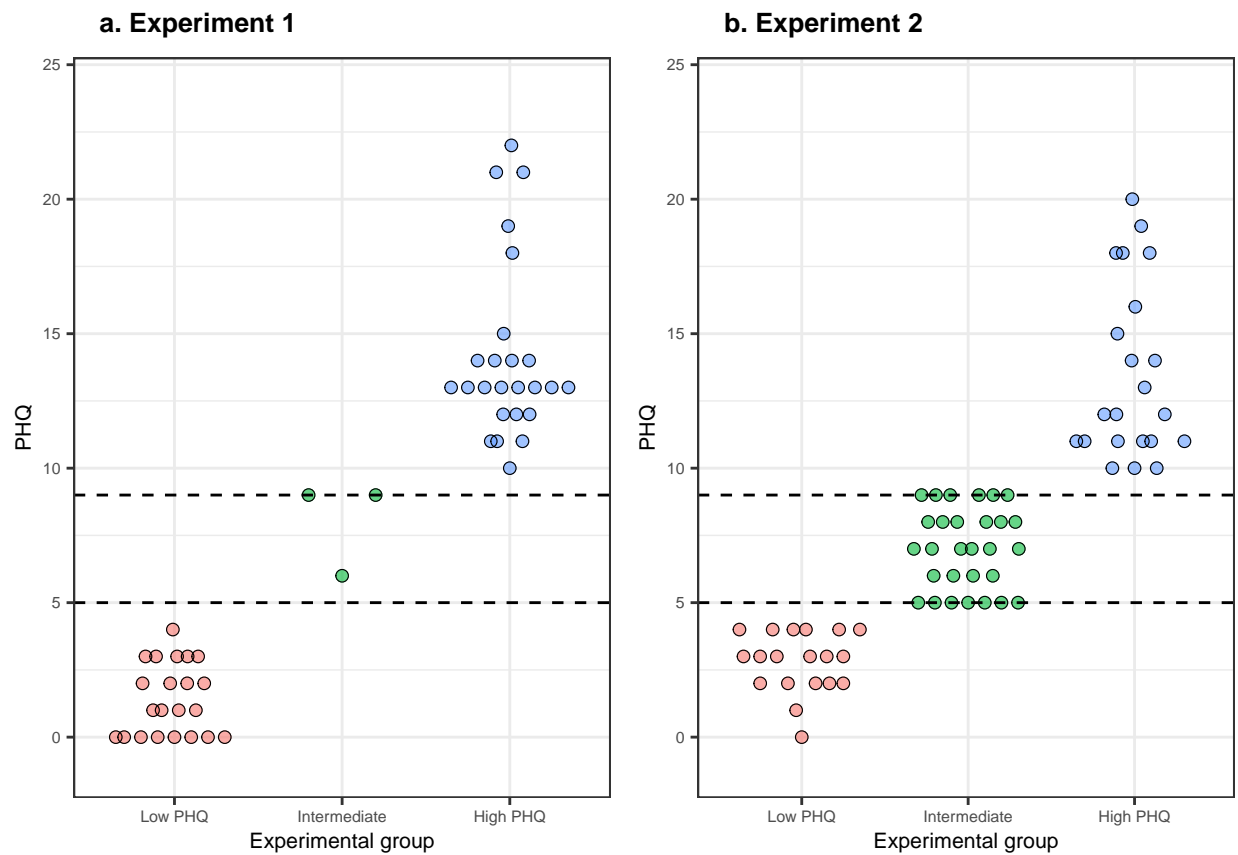
```

PHQ = Pred$PHQ[1])

barplot_probe = barplot_probe +
  geom_path(data = InteractionDF_3, aes(x = x1, y = y1), size = 0.3) +
  geom_text(data = InteractionDF_3[1,], aes(x= x2, y = y2, label = asterisk), size = 5)
}
barplot_2_groups[[exp_i]] = barplot_probe
}

# Print and save plots ----
### PHQ Distribution ----
PHQ_ditribution_plot_merged = ggarrange(PHQ_ditribution_plot[[1]], PHQ_ditribution_plot[[2]],
  font.label = list(size = 10),
  labels = c("a. Experiment 1", "b. Experiment 2"),
  common.legend = TRUE, ncol = 2, nrow = 1, legend = "bottom")
PHQ_ditribution_plot_merged

```



```

save_plot(PHQ_ditribution_plot_merged, file=paste0(figures_path,"PHQ_Distributions"), width = 5, height

```

```

## pdf
## 2

```

```

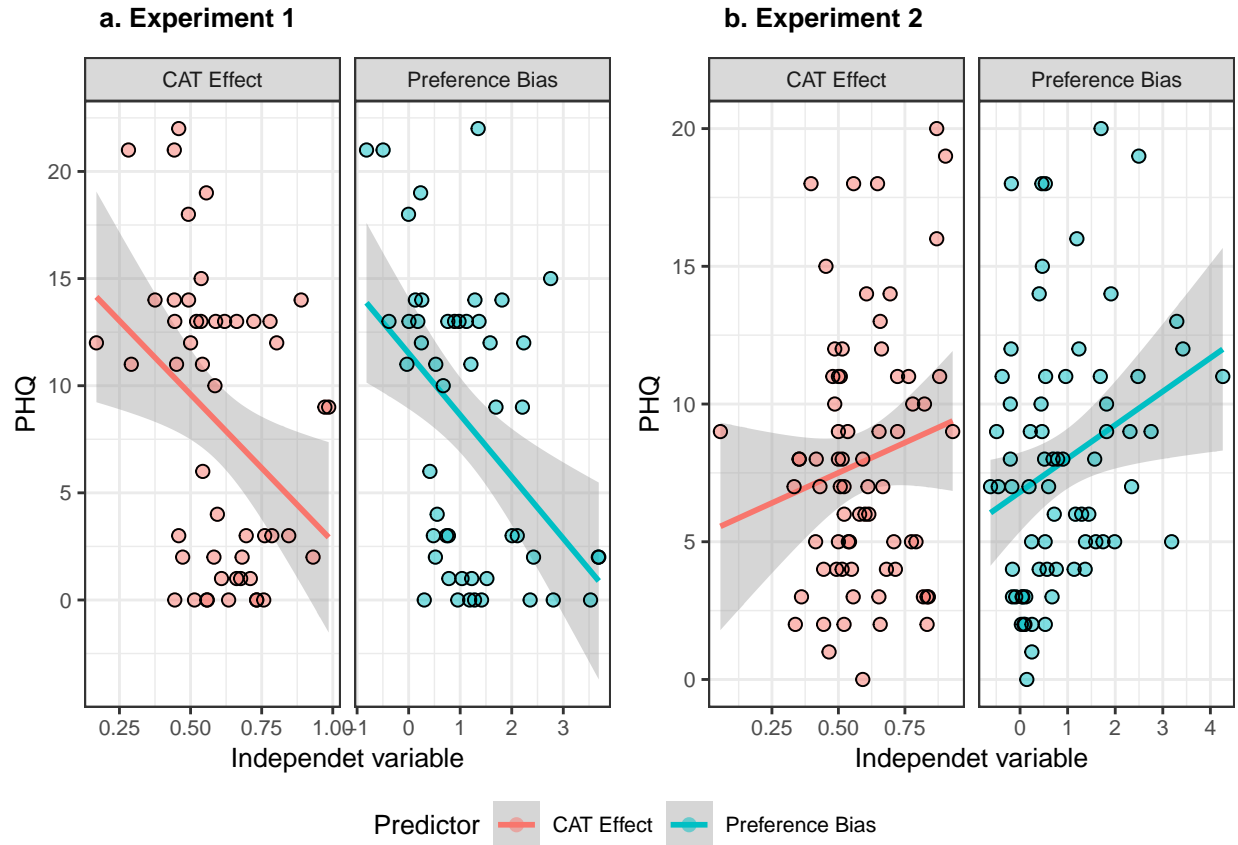
### PHQ prediction model ----
PHQ_prediction_merged = ggarrange(scatterplot_PHQ_prediction[[1]], scatterplot_PHQ_prediction[[2]],

```



```
PHQ_prediction_merged
```

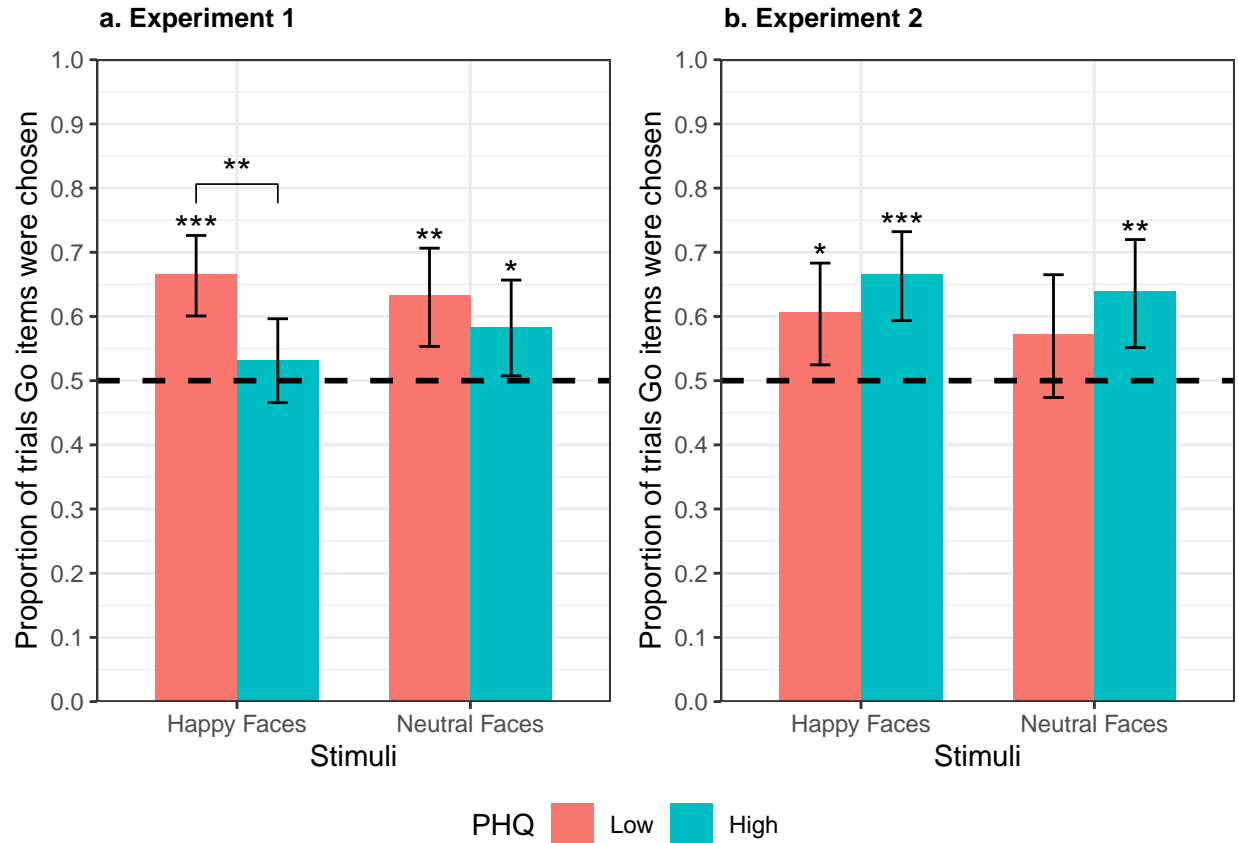
```
font.label = list(size = 10),
labels = c("a. Experiment 1", "b. Experiment 2"),
common.legend = TRUE, ncol = 2, nrow = 1, legend = "bottom")
```



```
save_plot(PHQ_prediction_merged, file=paste0(figures_path,"PHQ_Prediction"), width = 7, height = 3)
```

```
## pdf
## 2
```

```
### Probe ----
barplot_2_groups_merged = ggarrange(barplot_2_groups[[1]], barplot_2_groups[[2]],
font.label = list(size = 10),
labels = c("a. Experiment 1", "b. Experiment 2"),
common.legend = TRUE, ncol = 2, nrow = 1, legend = "bottom")
barplot_2_groups_merged
```



```
save_plot(barplot_2_groups_merged, file=paste0(figures_path,"ProbeResults"), width = 7, height = 4)
```

```
## pdf
## 2
```

```
# Summarize ----
### Probe ----
for (exp_i in 1:2) {
  cat("\nExperiment", exp_i, "\n=====\n")
  model_summary_i = model_summary_both[[exp_i]]
  for (effect_i in 1:nrow(model_summary_i)){
    eff = model_summary_i[effect_i,]
    p = round(eff$p,3)
    if (p<0.005){
      p = eff$p
    }
    cat(eff$Effect, ": prop. = ", round(eff$prop*100,2), "%, OR = ", round(eff$OR,2),
        ", 95% CI [", round(eff$OR_lower,2), ", ", round(eff$OR_upper,2), "], p = ", p, "\n",
        sep = "")
  }
}
```

```
##
## Experiment 1
## =====
```

```
## HappyFaces LowPHQ: prop. = 66.64%, OR = 2, 95% CI [1.5, 2.65], p = 1.698667e-06
## HappyFaces HighPHQ: prop. = 53.16%, OR = 1.14, 95% CI [0.87, 1.48], p = 0.347
## NeutralFaces LowPHQ: prop. = 63.32%, OR = 1.73, 95% CI [1.24, 2.41], p = 0.001254518
## NeutralFaces HighPHQ: prop. = 58.4%, OR = 1.4, 95% CI [1.03, 1.91], p = 0.032
## HappyFaces: Low PHQ > High PHQ: prop. = 63.77%, OR = 1.76, 95% CI [1.19, 2.59], p = 0.00421575
## NeutralFaces: Low PHQ > High PHQ: prop. = 55.15%, OR = 1.23, 95% CI [0.78, 1.94], p = 0.371
## Interaction: prop. = 58.87%, OR = 1.43, 95% CI [0.92, 2.22], p = 0.111
##
## Experiment 2
## =====
## HappyFaces LowPHQ: prop. = 60.67%, OR = 1.54, 95% CI [1.1, 2.16], p = 0.011
## HappyFaces HighPHQ: prop. = 66.64%, OR = 2, 95% CI [1.46, 2.73], p = 1.488614e-05
## NeutralFaces LowPHQ: prop. = 57.21%, OR = 1.34, 95% CI [0.9, 1.99], p = 0.15
## NeutralFaces HighPHQ: prop. = 63.99%, OR = 1.78, 95% CI [1.23, 2.57], p = 0.002221925
## HappyFaces: Low PHQ > High PHQ: prop. = 43.57%, OR = 0.77, 95% CI [0.49, 1.22], p = 0.269
## NeutralFaces: Low PHQ > High PHQ: prop. = 42.94%, OR = 0.75, 95% CI [0.44, 1.29], p = 0.302
## Interaction: prop. = 50.64%, OR = 1.03, 95% CI [0.61, 1.72], p = 0.923
```

PHQ Prediction ----

```
for (exp_i in 1:2) {
  model_full = model_PHQ_predict_full[[exp_i]] %>% summary()
  model_main = model_PHQ_predict_main[[exp_i]] %>% summary()

  model_summary_full = model_full$coefficients %>% as.data.frame() %>% mutate(df = model_full$df[2])
  model_summary_main = model_main$coefficients %>% as.data.frame() %>% mutate(df = model_main$df[2])
  model_summary_i = model_summary_main[2:3,] %>% rbind(model_summary_full[4,])
  colnames(model_summary_i) = c("beta", "SE", "t", "p", "df")
  model_summary_i = model_summary_i %>%
    mutate(Effect = c("CAT effect", "Preference bias", "Interaction"),
           CI_upper = beta + qt(p=0.975, df = df)*SE,
           CI_lower = beta + qt(p=0.025, df = df)*SE
    )
  cat("\nExperiment", exp_i, "\n=====\n")

  for (effect_i in 1:nrow(model_summary_i)){
    eff = model_summary_i[effect_i,]
    p = round(eff$p, 3)
    if (p < 0.005){
      p = eff$p
    }
    cat(eff$Effect, ": \u03b2 = ", round(eff$beta, 2), # \u03b2 = beta unicode
        ", 95% CI [", round(eff$CI_lower, 2), ", ", round(eff$CI_upper, 2), "], t(",
        eff$df, ") = ", round(eff$t, 2), ", p = ", p, "\n",
        sep = "")
  }
}
```

```
##
## Experiment 1
## =====
## CAT effect: <U+03B2> = -0.25, 95% CI [-0.51, 0.02], t(47) = -1.89, p = 0.065
## Preference bias: <U+03B2> = -0.38, 95% CI [-0.64, -0.11], t(47) = -2.88, p = 0.006
## Interaction: <U+03B2> = 0.19, 95% CI [-0.05, 0.42], t(46) = 1.6, p = 0.116
##
```

```
## Experiment 2
## =====
## CAT effect:  $\langle U+03B2 \rangle = 0.09$ , 95% CI  $[-0.15, 0.33]$ ,  $t(67) = 0.75$ ,  $p = 0.459$ 
## Preference bias:  $\langle U+03B2 \rangle = 0.25$ , 95% CI  $[0.01, 0.49]$ ,  $t(67) = 2.04$ ,  $p = 0.046$ 
## Interaction:  $\langle U+03B2 \rangle = 0.24$ , 95% CI  $[-0.03, 0.51]$ ,  $t(66) = 1.79$ ,  $p = 0.078$ 
```