```
#Remove any existing objects
rm (list = ls())
#Load required packages
library(readxl)
library(dplyr)
library(rmcorr)
library(lme4)
library(lmerTest)
library(performance)
library(ggplot2)
library(ggpubr)
library(stan)
library(brms)
library(loo)

#Load and check data from Agree group and Conflict group
setwd("C:/Users/user/Documents/Zebra Finch/Demonstrator Project")
Agree <- read_excel("Agree_Data.xlsx", na = "na")
Conflict <- read_excel("Conflict_Data.xlsx", na = "na")
View(Agree)
View(Conflict)
#Calculate descriptive stats for the Agree group
summary_agree <- Agree %>%
  summary()
summary_agree
#Calculate descriptive stats for the Conflict group
summary_conflict <- Conflict %>%
  summary()
summary_conflict
```

```
####


###Data Key###
#Pair_ID = pair id
#Group = treatment group (agree or conflict)
#Initial_Time = number of sessions in the initial preference test (ITP)
#PreObs_Time = trial duration of pre-observation phase in hours
#Obs_Time = trial duration of observation phase in hours
#Final_Time = trial duration of final preference test (FPT) in hours
#I_Pref_Prop = Proportion of time with preferred color in IPT
#I_Demo_Prop = Proportion of time with demonstrated color in IPT
#I_OptOut_Prop = Proportion of time with neither preferred nor demonstrated color in IPT
#I_Social_Prop = Proportion of time with socially demonstrated color in IPT
#I_Asocial_Prop = Proportion of time with asocial color in IPT
#F_Preferred = Number of deposits made using preferred color in FPT
#F_Demonstrated = Number of deposits made using demonstrated color in FPT
#F_OptOut = Number of deposits made using neither preferred nor demonstrated color in FPT\
#F_Social = Number of deposits made using socially demonstrated color in FPT
#F_Asocial = Number of deposits made using asocial color in FPT
#F_Pref_Prop = Proportion of time with preferred color in FPT
#F_Demo_Prop = Proportion of time with demonstrated color in FPT
#F_OptOut_Prop = Proportion of time with neither preferred nor demonstrated color in FPT
#F_Social_Prop = Proportion of time with socially demonstrated color in FPT
#F_Asocial_Prop = Proportion of time with asocial color in FPT
####


###Does duration of pre-observation trials differ between the two groups?###
#Check distribution of pre-observation test times in each group
hist(Conflict$PreObs_Time)
```

```
shapiro.test(Conflict$PreObs_Time) #Not normal
hist(Agree$PreObs_Time)
shapiro.test(Agree$PreObs_Time) #Not normal
#Perform Mann-Whitney U test to compare trial duration in each group
#First make data frame with pre-observation trial time for each group, where Conflict = C and Agree = A
Pre_trial.time <- c(Conflict$PreObs_Time, Agree$PreObs_Time)
trial.group <- factor(c("C", "C", "C", "C", "C", "C", "C", "C", "C", "C", "C", "C", "C", "C", "C",
               "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A"))
MWUtest.data <- data.frame(trial.group, Pre_trial.time)
#Perform Mann-Whitney U test
wilcox.test(Pre_trial.time ~ trial.group, data = MWUtest.data, na.rm = TRUE, paired = FALSE, exact = FALSE, conf.int = TRUE)
#No significant difference in pre-observation trial duration between groups
#Calculate standard deviations for pre-observation trial duration in each group
sd_agree <- Agree %>%
  summarize(sd.PreObs = sd(PreObs_Time))
sd_agree
sd_conflict <- Conflict %>%
  summarize(sd.PreObs = sd(PreObs_Time))
sd_conflict
####


###Does duration of observation trials differ between the two groups?###
#Check distribution of observation test times
hist(Conflict$Obs_Time)
shapiro.test(Conflict$Obs_Time) #Normal
hist(Agree$Obs_Time)
shapiro.test(Agree$Obs_Time) #Not normal
#Perform Mann-Whitney U test to compare trial duration in each group
```

```r
#First make data frame for test with observation trial times

Obs_Trial.time <- c(Conflict$Obs_Time, Agree$Obs_Time)

MWUtest2.data <- data.frame(trial.group, Obs_Trial.time)

#Perfrom Mann-Whitney U Test

wilcox.test(Obs_Trial.time ~ trial.group, data = MWUtest2.data, na.rm = TRUE, paired = FALSE, exact = FALSE, conf.int = TRUE)

#No significant difference in observation trial duration between groups

#Calculate standard deviations for observation trial lengths in each group

sd_agree <- Agree %>%

  summarize(sd.Obs = sd(Obs_Time))

sd_agree

sd_conflict <- Conflict %>%

  summarize(sd.Obs = sd(Obs_Time))

sd_conflict

####


###Does  initial preference strength differ between in each group?###

#First make data frame with pre-observation trial time for each group, where Conflict = C and Agree = A

Pref_Strength <- c(Conflict$I_Pref_Prop, Agree$I_Pref_Prop)

MWUtest3.data <- data.frame(trial.group, Pref_Strength)

#Perform Mann-Whitney U test

wilcox.test(Pref_Strength ~ trial.group, data = MWUtest3.data, na.rm = TRUE, paired = FALSE, exact = FALSE, conf.int = TRUE)

#No significant difference in pre-observation trial duration between groups



###Does Total Interaction Time with a color correlate with other measures of preference?###

#Load data and subset for the Initial Preference Test

Preference_Data <- read_excel("Preference_Test_Scores.xlsx", na = "na")
```

```
IPT_Data <- Preference_Data %>%
  filter(Test == "Initial")
####
###Data Key###
#Pair ID = pair id
#Group = treatment group (agree or conflict)
#Test = preference test (initial or final)
#Color_Rank = whether color was 1st, 2nd, or 3rd preferred in the test
#Color = pink, yellow, green
#Color_Info = Associated information with the color. Agree group (initially preferred, demonstrated, neutral) or Conflict group (initially preferred, social, asocial)
#Proportion_Score = proportion of total color interaction time spent with color (initial preference test) or proportion of color in first 20 nest deposits(final preference test)
#Total_Time = Time spent with color in initial preference test
#No_Touches = Number of touches in initial preference test
#Touch_Latency = Time latency to first touch color in intial preference test
#No_Deposits = Number of times color was deposited into nest during final preference test
####

###Evaluate behavioural measures###
#Run repeated measures correlation between total interation time and number of touches.
rmcorr(participant = Pair_ID, Total_Time, No_Touches, data = IPT_Data, CI.level = 0.95, CIs = "analytic")
#Run repeated measures correlation between total interation time and latency to touch.
rmcorr(participant = Pair_ID, Total_Time, Touch_Latency, data = IPT_Data, CI.level = 0.95, CIs = "analytic")
#Is there a relationship between interaction time and preference rank?
preference_rank.mod <- lmer(Total_Time ~ Color_Rank + (1|Pair_ID), data = IPT_Data)
#check model assumptions
check_model(preference_rank.mod)
#transform total time and rerun model
```

```
IPT_Data$log.Total_Time <- log(IPT_Data$Total_Time + 1)
preference_rank.mod2 <- lmer (log.Total_Time ~ Color_Rank + (1|Pair_ID), data = IPT_Data)
#check model assumptions
check_model(preference_rank.mod2)
summary(preference_rank.mod2) #Significant relationship
####

###Summary statistics for each color rank###
rank_summary <- Preference_Data %>%
  group_by(Color_Rank) %>%
  summarise(meanscore = mean(Proportion_Score),
            sdscore = sd(Proportion_Score),
            minscore = min(Proportion_Score),
            maxscore= max(Proportion_Score))
rank_summary
####

###Does preference score chance between initial a final preference tests?###
#Wilcoxon tests comparing preference score for each color type before and after observation
#In the Agree group...
coin::wilcoxsign_test(Agree$I_Pref_Prop ~ Agree$F_Pref_Prop, exact = FALSE, paired = TRUE)#change in initially preferred
coin::wilcoxsign_test(Agree$I_Demo_Prop ~ Agree$F_Demo_Prop, exact = FALSE, paired = TRUE)#no change in demonstrated
coin::wilcoxsign_test(Agree$I_OptOut_Prop ~ Agree$F_OptOut_Prop, exact = FALSE, paired = TRUE)#no change in "opt out"
#Calculate standard deviations for each color in initial and final preference tests
sd_agree <- Agree %>%
  summarize(sdI.pref = sd(I_Pref_Prop),
            sdF.pref = sd(F_Pref_Prop),
          sdI.demo = sd(I_Demo_Prop),
          sdF.demo = sd(F_Demo_Prop),
          sdI.opt = sd(I_OptOut_Prop),
          sdF.opt = sd(F_OptOut_Prop))
sd_agree
#In the Conflict group...
coin::wilcoxsign_test(Conflict$I_Pref_Prop ~ Conflict$F_Pref_Prop, exact = FALSE, paired = TRUE)#change in initially preferred
coin::wilcoxsign_test(Conflict$I_Social_Prop ~ Conflict$F_Soc_Prop, exact = FALSE, paired = TRUE)#change in social color
coin::wilcoxsign_test(Conflict$I_Asocial_Prop ~ Conflict$F_Asoc_Prop, exact = FALSE, paired = TRUE)#no change in asocial color
#Calculate standard deviations for each color in initial and final preference tests
sd_conflict <- Conflict %>%
  summarize(sdI.pref = sd(I_Pref_Prop),
          sdF.pref = sd(F_Pref_Prop),
          sdI.soc = sd(I_Social_Prop),
          sdF.soc = sd(F_Soc_Prop),
          sdI.asoc = sd(I_Asocial_Prop),
          sdF.asco = sd(F_Asoc_Prop))
sd_conflict
####


###Make plots to vizualise changes between IPT and FPT###
#Cleanup function to remove grids
cleanup = theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
             panel.background = element_blank(), axis.line = element_line(color = "black"))
#Make dataframe for Agree group plot
Agree_Plot_Data <- Preference_Data %>%
  filter(Group == "Agree")
Agree_Plot_Data$Test <- factor(Agree_Plot_Data$Test, levels = c("Initial", "Final"))
Agree_Plot_Data$Color_Info <- factor(Agree_Plot_Data$Color_Info, levels = c("Preferred", "Demonstrated", "Neutral"))
#code plot
Agree_Plot <- ggplot(Agree_Plot_Data, aes(x = Test, y = Proportion_Score, fill = Test))+
  geom_boxplot()+
  geom_point(position = position_dodge(width = 0.75))+
  geom_line(aes(group = Pair_ID), linetype = "dashed")+
  ylab("Preference score (%)")+
  xlab("Color (Agree group)")+
  cleanup+
  theme(text = element_text(size = 17), axis.text.x = element_blank())+
  facet_wrap(~Color_Info, strip.position = "bottom" )
Agree_Plot
#Make dataframe for Conflict group plot
Conflict_Plot_Data <- Preference_Data %>%
  filter(Group == "Conflict")
Conflict_Plot_Data$Test <- factor(Conflict_Plot_Data$Test, levels = c("Initial", "Final"))
Conflict_Plot_Data$Color_Info <- factor(Conflict_Plot_Data$Color_Info, levels = c("Preferred", "Social", "Asocial"))
#code plot
Conflict_Plot <- ggplot(Conflict_Plot_Data, aes(x = Test, y = Proportion_Score, fill = Test))+
  geom_boxplot()+
  geom_point(position = position_dodge(width = 0.75))+
  geom_line(aes(group = Pair_ID), linetype = "dashed")+
  ylab("Preference score (%)")+
  xlab("Color (Conflict group)")+
  cleanup +
  theme(text = element_text(size = 17), axis.text.x = element_blank())+
  facet_wrap(~Color_Info, strip.position = "bottom" )
Conflict_Plot
#Combine plots
Figure3<-ggarrange(Agree_Plot, Conflict_Plot,
                   labels = c("(a)", "(b)"),
                   font.label = list(size = 12),
                   ncol = 1, nrow = 2,
                   common.legend = TRUE, legend = "bottom")
Figure3
####

###Does initial bias influence social information use?###
#First fit the model
Strength <- read_excel("Strength.xlsx", na = "na")
strength_mod<-glm(F_Social ~ I_Pref_Prop + Group, family = poisson, data = Strength)
#Check for overdispersion
check_overdispersion(strength_mod) #Overdispersion detected, refit with quasi-poisson distribution
strength_mod<-glm(F_Social ~ I_Pref_Prop, family = quasipoisson, data = Strength)
summary(strength_mod)
####

###Determine likelihood of a color being chose in the Final preference test given strength of initial bias###
#Run this code if stan is not working
remove.packages(c("rstan","StanHeaders"))
install.packages("StanHeaders", repos = c("https://mc-stan.org/r-packages/", getOption("repos")))
install.packages("rstan", repos = c("https://mc-stan.org/r-packages/", getOption("repos")))
library(rstan)
```

```
#Correct 0 values in order to run models using Dirichlet distribution
Agree$c.F_Pref_Prop <- (Agree$F_Pref_Prop*13 + 2) / 14
Agree$c.F_Demo_Prop <- (Agree$F_Demo_Prop*13 + 2) / 14
Agree$c.F_Opt_Prop <- (Agree$F_OptOut_Prop*13 + 2) / 14
#Create dataframe for Bayesian analysis of Agree group
Agree_df <- data.frame(
  Pref = Agree$c.F_Pref_Prop, Demo = Agree$c.F_Demo_Prop, Opt = Agree$c.F_Opt_Prop,
  Pref_Strength = Agree$I_Pref_Prop
) %>%
  mutate(
    size = Pref + Demo + Opt,
    Pref = Pref / size,
    Demo = Demo / size,
    Opt = Opt / size
  )
Agree_df
#obtain priors to run the model
bind <- function(...) cbind(...)
prior <- get_prior(bind(Pref, Demo, Opt) ~ Pref_Strength, data = Agree_df, family = "dirichlet")
prior
#write code for model to run using stan language
make_stancode(bind(Pref, Demo, Opt) ~ Pref_Strength, Agree_df, prior = prior, dirichlet())
make_standata(bind(Pref, Demo, Opt) ~ Pref_Strength, Agree_df, prior = prior, dirichlet())
#run model
Agree_mod <- brm(bind(Pref, Demo, Opt) ~ Pref_Strength, Agree_df, family = "dirichlet", prior = prior, chains = 4, iter = 20000, warmup = 5000)
summary(Agree_mod, prob = 0.95) #wide, overlapping 95%CI
#Check model diagnostics
plot(Agree_mod)
```

```
#Extract and plot conditional effects
C_eff <- conditional_effects(Agree_mod, categorical = TRUE)
C_eff
#Plot effects of model
Bayes_plot1 <- plot(C_eff, plot = FALSE) [[1]] +
  xlab("Bias strength (%) towards preferred colour") +
  ylab("Likelihood (%) of being selected in final test") +
  theme(text = element_text(size = 17))+
  scale_x_continuous(breaks = c(0.2, 0.4, 0.6, 0.8, 1), labels = c("20", "40", "60", "80", "100"))+
  scale_y_continuous(breaks = c(0.2, 0.4, 0.6, 0.8, 1), labels = c("20", "40", "60", "80", "100"))+
  scale_color_manual('Agree group:', labels = c("Preferred", "Demonstrated", "Neutral"), values
=c("#00AFBB", "#C4961A", "#FC4E07"))+
  scale_fill_manual('Agree group:', labels = c("Preferred", "Demonstrated", "Neutral"), values =
c("#00AFBB", "#C4961A", "#FC4E07"))+
  theme(legend.position = "top", text = element_text(size = 20))+
  cleanup
Bayes_plot1
####


##################
#In Conflict Group...
#0 value correction
Conflict$c.F_Pref_Prop <- (Conflict$F_Pref_Prop*14 + 2) / 15
Conflict$c.F_Soc_Prop <- (Conflict$F_Soc_Prop*14 + 2) / 15
Conflict$c.F_Asoc_Prop <- (Conflict$F_Asoc_Prop*14 + 2) / 15
#Bayesian analysis - Conflict
Conflict_df <- data.frame(
  Pref = Conflict$c.F_Pref_Prop, Soc = Conflict$c.F_Soc_Prop, Asoc = Conflict$c.F_Asoc_Prop,
  Pref_Strength = Conflict$I_Pref_Prop
) %>%
  mutate(
    size = Pref + Soc + Asoc,
    Pref = Pref / size,
    Soc = Soc / size,
    Asoc = Asoc / size
  )
Conflict_df
#obtain priors to run the model
prior <- get_prior(bind(Pref, Soc, Asoc) ~ Pref_Strength, data = Conflict_df, family = "dirichlet")
prior
#write code for model to run using stan language
make_stancode(bind(Pref, Soc, Asoc) ~ Pref_Strength, Conflict_df, prior = prior, dirichlet())
make_standata(bind(Pref, Soc, Asoc) ~ Pref_Strength, Conflict_df, prior = prior, dirichlet())
#run model
Conflict_mod <- brm(bind(Pref, Soc, Asoc) ~ Pref_Strength, Conflict_df, family = "dirichlet", prior = prior, chains = 4, iter = 20000, warmup = 5000)
summary(Conflict_mod, prob = 0.95)
#Check model diagnostics
plot(Conflict_mod)
#Extract and plot conditional effects
C_eff <- conditional_effects(Conflict_mod, categorical = TRUE)
C_eff
#Plot effects of model
Bayes_plot2 <- plot(C_eff, plot = FALSE) [[1]] +
  xlab("Bias strength (%) towards preferred colour") +
  ylab("Likelihood (%) of being selected in final test") +
  theme(text = element_text(size = 17))+
  scale_x_continuous(breaks = c(0.2, 0.4, 0.6, 0.8, 1), labels = c("20", "40", "60", "80", "100"))+
  scale_y_continuous(breaks = c(0.2, 0.4, 0.6, 0.8, 1), labels = c("20", "40", "60", "80", "100"))+
  scale_color_manual('Conflict group:', labels = c("Preferred", "Social", "Asocial"), values = c( "#00AFBB","#CC79A7","forestgreen"))+
  scale_fill_manual('Conflict group:', labels = c("Preferred", "Social", "Asocial"), values = c( "#00AFBB","#CC79A7","forestgreen"))+
  theme(legend.position = "top", text = element_text(size = 20))+
  cleanup
Bayes_plot2
#combine plots
Figure4<-ggarrange(Bayes_plot1, Bayes_plot2,
                   labels = c("(a)", "(b)"),
                   font.label = list(size =18),
                   ncol = 2, nrow = 1)
Figure4
####



###Supplementary material###
#Make a stacked bar chart to show color preferences
Stacked <- read_excel("Stacked_Plot_Data.xlsx")
View(Stacked)
Stacked$ID <- as.numeric (Stacked$ID)
#Subset dataframes for each group
Stacked_Agree <- Stacked %>%
  filter(Group == "Agree")
Stacked_Conflict <- Stacked %>%
  filter(Group == "Conflict")
```

```
#Make plot for agree group
sm1<-ggplot(data = Stacked_Agree, aes(fill = Colour, y = Prop, x = ID))+
  geom_bar(stat = "identity")+
  scale_fill_manual(values=c("green4","hotpink","yellow2"), name = "Material color")+
  ylab("Preference (%)")+
  xlab("Males")+
  theme(legend.position = "bottom", axis.text.x = element_blank(), axis.ticks.x = element_blank(), text =
element_text(size = 25))+
  cleanup
sm1

sm2<-ggplot(data = Stacked_Conflict, aes(fill = Colour, y = Prop, x = ID))+
  geom_bar(stat = "identity")+
  scale_fill_manual(values=c("green4","hotpink","yellow2"), name = "Material color")+
  ylab("Preference (%)")+
  xlab("Males")+
  theme(legend.position = "none", axis.text.x = element_blank(), axis.ticks.x = element_blank(), text =
element_text(size = 25))+
  cleanup
sm2
#Combine plots
SMFigure<-ggarrange(sm1, sm2,
                    labels = c("Agree", "Conflict"),
                    font.label = list(size = 25),
                    ncol = 2, nrow = 1, common.legend = TRUE, label.x=0.4, label.y = 1.015)
SMFigure
####
```

```
###Rerun analyses excluding outliers with weak initial bias###
Conflict_OUT <- read_excel("Conflict_Outlier_Data.xlsx")
View(Conflict_OUT)
#0 value correction
Conflict_OUT$c.F_Pref_Prop <- (Conflict_OUT$F_Pref_Prop*14 + 2) / 15
Conflict_OUT$c.F_Soc_Prop <- (Conflict_OUT$F_Soc_Prop*14 + 2) / 15
Conflict_OUT$c.F_Asoc_Prop <- (Conflict_OUT$F_Asoc_Prop*14 + 2) / 15
#Bayesian analysis - Conflict
Conflict_OUT_df <- data.frame(
  Pref = Conflict_OUT$c.F_Pref_Prop, Soc = Conflict_OUT$c.F_Soc_Prop, Asoc =
Conflict_OUT$c.F_Asoc_Prop,
  Pref_Strength = Conflict_OUT$I_Pref_Prop
) %>%
  mutate(
    size = Pref + Soc + Asoc,
    Pref = Pref / size,
    Soc = Soc / size,
    Asoc = Asoc / size
  )
Conflict_OUT_df
#obtain priors to run the model
prior <- get_prior(bind(Pref, Soc, Asoc) ~ Pref_Strength, data = Conflict_OUT_df, family = "dirichlet")
prior
#write code for model to run using stan language
make_stancode(bind(Pref, Soc, Asoc) ~ Pref_Strength, Conflict_OUT_df, prior = prior, dirichlet())
make_standata(bind(Pref, Soc, Asoc) ~ Pref_Strength, Conflict_OUT_df, prior = prior, dirichlet())
#run model
Conflict_OUT_mod <- brm(bind(Pref, Soc, Asoc) ~ Pref_Strength, Conflict_OUT_df, family = "dirichlet",
prior = prior, chains = 4, iter = 20000, warmup = 5000)
```

```
summary(Conflict_OUT_mod, prob = 0.95)
#Check model diagnostics
plot(Conflict_OUT_mod)
#Extract and plot conditional effects
C_eff <- conditional_effects(Conflict_OUT_mod, categorical = TRUE)
C_eff
#Plot effects of model
Bayes_plot3 <- plot(C_eff, plot = FALSE) [[1]] +
  xlab("Bias strength (%) towards preferred colour") +
  ylab("Likelihood (%) of being selected in final test") +
  theme(text = element_text(size = 17))+
  scale_x_continuous(breaks = c(0.2, 0.4, 0.6, 0.8, 1), labels = c("20", "40", "60", "80", "100"))+
  scale_y_continuous(breaks = c(0.2, 0.4, 0.6, 0.8, 1), labels = c("20", "40", "60", "80", "100"))+
  scale_color_manual('Conflict group:', labels = c("Preferred", "Social", "Asocial"), values = c(
"#00AFBB","#CC79A7","forestgreen"))+
  scale_fill_manual('Conflict group:', labels = c("Preferred", "Social", "Asocial"), values = c(
"#00AFBB","#CC79A7","forestgreen"))+
  theme(legend.position = "top", text = element_text(size = 20))+
  cleanup
Bayes_plot3
```