Recognition Plots

Gözem Turan

29 6 2022

library(lme4)  
library(car)  
library(tidyr)  
library(Rmisc)  
library(rstatix)  
library(dplyr)  
library(ggpubr)  
library(ggplot2)  
library(ez)  
library(psycho)  
library(sjPlot)  
library(PupillometryR)  
library(MetBrewer)  
library(here)

# Open data seperately

# setting relative path  
here\_path <- here()  
  
# relative data path  
data\_path <- file.path(here\_path, "data", "raw\_data")  
  
# reading the data in  
df\_m <- read.csv(file = file.path(data\_path, "m\_part3\_rt\_out.csv"))  
df\_w <- read.csv(file = file.path(data\_path, "w\_part3\_rt\_out.csv"))

Preparations for df\_m

# setting pe as a factor  
df\_m <- df\_m %>%   
 mutate(pe = factor(pe, levels = c("lowPE", "medPE", "highPE")))  
  
# setting participant to a factor variable  
df\_m <- df\_m %>%   
 mutate(participant = as.factor(participant))  
  
# setting confidence to a factor variable  
df\_m <- df\_m %>%   
 mutate(conf = factor(conf, levels = c("1", "2", "3", "4")))  
  
# removing NA values from the dataframe  
df\_m <- df\_m %>%   
 na.omit()  
  
# creating column with all values as "Experiment 2"  
df\_m <- df\_m %>%   
 mutate(study = "Experiment 1")

# preparations for df\_w

# change the naming for PE levels to be consistend  
df\_w <- df\_w %>%  
 mutate(pe = recode(PE\_level, low = "lowPE", medium = "medPE", high = "highPE"))  
  
# converting pe to a factor variable  
df\_w <- df\_w %>%   
 mutate(pe = factor(pe, levels = c("lowPE", "medPE", "highPE"))) # pe  
  
  
# setting participant to a factor variable  
df\_w <- df\_w %>%   
 mutate(participant = factor(participant)) # participants  
  
# setting confidence to a factor variable  
df\_w <- df\_w %>%   
 mutate(conf = factor(conf, levels = c("1", "2", "3", "4"))) # confidence  
  
# removing NA values from the dataframe  
df\_w <- df\_w %>%   
 na.omit()  
  
# creating column with all values as "Experiment 2"  
df\_w <- df\_w %>%   
 mutate(study = "Experiment 2")

# Combine two data sets

# combining the two dataframes   
# will set NA values to any columns that one dataset does not have  
df <- bind\_rows(df\_m, df\_w)  
# just ignore the na values  
  
# setting study as a factor variable  
df <- df %>%   
 mutate(study = as.factor(study))

# Aggregated data

# creates a summary dataframe of the corrAns varible (count, mean, sd, se, ci)  
# grouped by pe and study  
df\_corr\_ans <- df %>%   
 group\_by(pe, study) %>%   
 summarise(  
 N = n(),  
 mean\_corr\_ans = mean(corrAns, na.rm = TRUE),  
 sd = sd(corrAns, na.rm = TRUE),  
 se = sd / sqrt(N)  
 )

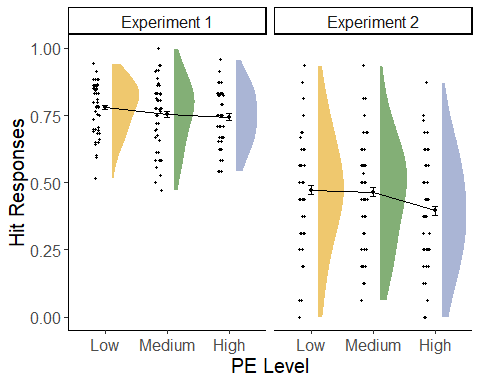
## `summarise()` has grouped output by 'pe'. You can override using the `.groups`  
## argument.

# creates a summary dataframe of the corrAns varible (count, mean, sd, se, ci)  
# grouped by pe, participant, and study  
df\_corr\_ans\_part <- df %>%   
 group\_by(pe, participant, study) %>%   
 summarise(  
 N = n(),  
 mean\_corr\_ans = mean(corrAns, na.rm = TRUE),  
 sd = sd(corrAns, na.rm = TRUE),  
 se = sd / sqrt(N)  
 )

## `summarise()` has grouped output by 'pe', 'participant'. You can override using  
## the `.groups` argument.

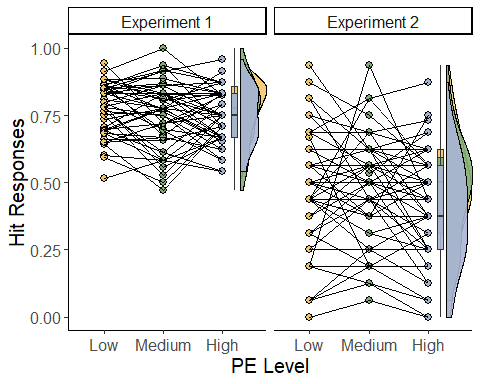
# First plot - Rainyclouds

# Violin plot of hit responses as a function of pe level, for Experiments 1 and 2  
ggplot(df\_corr\_ans\_part, aes(x = pe, y = mean\_corr\_ans, fill = pe)) +  
 geom\_flat\_violin(position = position\_nudge(x = .1, y = 0), adjust = 1.5, trim = TRUE, color = "white") +  
 geom\_point(aes(x = as.numeric(pe) - .15, y = mean\_corr\_ans), position = position\_jitter(width = .05, height = 0), size = 1, shape = 20) +  
 # box plot  
 # geom\_boxplot(aes(x = pe, y = mean\_corr\_ans),outlier.shape = NA, width = .1, colour = "black") +  
 geom\_errorbar(data = df\_corr\_ans, aes(ymin = mean\_corr\_ans - se, ymax = mean\_corr\_ans + se, width = 0.1)) +  
 geom\_point(data = df\_corr\_ans, size = 1) +  
 geom\_line(data = df\_corr\_ans, aes(x = pe, y = mean\_corr\_ans), size = 0.5, group = 1) +  
 theme\_classic() +  
 scale\_fill\_manual(values = met.brewer("Derain", 5, type = "continuous")) +  
 scale\_color\_manual(values = met.brewer("Derain", 5, type = "continuous")) +  
 scale\_x\_discrete(name = "PE Level",   
 labels = c("lowPE" = "Low", "medPE" = "Medium", "highPE" = "High")) +  
 theme(  
 text = element\_text(family = "TT Times New Roman", size = 15),   
 legend.title = element\_text(size = 1),   
 legend.text = element\_text(size = 12),   
 legend.position = "None",   
 legend.direction = "vertical") +  
 ylab("Hit Responses") +  
 ylim(0, 1) +  
 # geom\_signif(comparisons = list(c("lowPE", "highPE")), map\_signif\_level=TRUE, y\_position = c(1.1, 1.2, 1.3)) +  
 facet\_wrap(. ~ study)



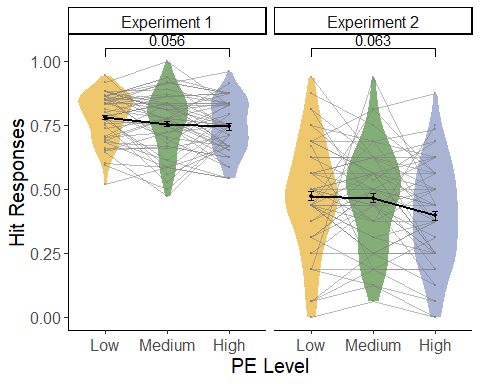
# Second plot - spaghetti & box & distribution

# Hit rate per person across different levels of pe, per each experiment  
ggplot(df\_corr\_ans\_part, aes(x = 1, y = mean\_corr\_ans, fill = pe)) +  
 geom\_flat\_violin(trim = TRUE, position = position\_nudge(x = 2.3), alpha = .9, color = "black") +  
 geom\_point(aes(x = factor(pe), y = mean\_corr\_ans, fill = factor(pe)), size = 2.5, shape = 21) +  
 geom\_line(aes(x = factor(pe), y = mean\_corr\_ans, group = participant)) +  
 geom\_boxplot(  
 width = .1, show.legend = FALSE,  
 outlier.shape = NA, alpha = .9,  
 position = position\_nudge(x = 2.2)  
 ) +  
 # geom\_line(data=df\_corr\_ans, aes(x=pe, y=mean\_corr\_ans), size=1.4, group=1, color = "pink") +  
 scale\_fill\_manual(values = met.brewer("Derain", 5, type = "continuous")) +  
 theme\_classic() +  
 scale\_x\_discrete(name = "PE Level", labels = c("lowPE" = "Low", "medPE" = "Medium", "highPE" = "High")) +  
 theme(  
 text = element\_text(family = "TT Times New Roman", size = 15),   
 legend.title = element\_text(size = 1),   
 legend.text = element\_text(size = 12),   
 legend.position = "None",   
 legend.direction = "vertical") +  
 ylab("Hit Responses") +  
 ylim(0, 1) +  
 # geom\_signif(comparisons = list(c("low", "dprimeMed"), c("dprimeLow", "dprimeHigh"), c("dprimeMed", "dprimeHigh")), map\_signif\_level=TRUE) +  
 facet\_wrap(. ~ study)



# Third plot - violin

# Plot of hit rate as a function of pe level, per each experiment  
# Significance shown for low vs. high pe level  
ggplot(df\_corr\_ans\_part, aes(x = pe, y = mean\_corr\_ans, fill = pe)) +  
 geom\_violin(color = "white", trim = TRUE) +  
 geom\_jitter(aes(x = pe, y = mean\_corr\_ans, group = participant, alpha = 0.5), color = "gray50", size = 0.5, position = position\_dodge(0)) +  
 geom\_line(aes(x = pe, y = mean\_corr\_ans, group = participant, alpha = 0.5), color = "gray50", position = position\_dodge(0)) +  
 geom\_point(data = df\_corr\_ans, size = 1) +  
 geom\_errorbar(data = df\_corr\_ans, aes(ymin = mean\_corr\_ans - se, ymax = mean\_corr\_ans + se, width = 0.1)) +  
 geom\_line(data = df\_corr\_ans, aes(x = pe, y = mean\_corr\_ans), size = 0.9, group = 1) +  
 scale\_x\_discrete(name = "PE Level", labels = c("lowPE" = "Low", "medPE" = "Medium", "highPE" = "High")) +  
 scale\_fill\_manual(values = met.brewer("Derain", 5, type = "continuous")) +  
 theme\_classic() +  
 theme(text = element\_text(family = "TT Times New Roman", size = 15), legend.title = element\_text(size = 1), legend.text = element\_text(size = 12), legend.position = "None", legend.direction = "vertical") +  
 ylab("Hit Responses") +  
 # ylim(0, 1) +  
 geom\_signif(comparisons = list(c("highPE", "lowPE")), map\_signif\_level = FALSE) +  
 facet\_wrap(. ~ study)



# CONFINDENCE

# Summary statistics for mean\_corr\_ans variable, grouped by pe, study, and conf  
df\_corr\_ans\_conf <- df %>%   
 group\_by(pe, study, conf) %>%   
 summarise(  
 N = n(),  
 mean\_corr\_ans = mean(corrAns, na.rm = TRUE),  
 sd = sd(corrAns, na.rm = TRUE),  
 se = sd / sqrt(N)  
 )

## `summarise()` has grouped output by 'pe', 'study'. You can override using the  
## `.groups` argument.

## Summary statistics for corrAns variable, grouped by pe, participant, study, and conf  
df\_corr\_ans\_all <- df %>%   
 group\_by(pe, participant, conf, study) %>%   
 summarise(  
 N = n(),  
 mean\_corr\_ans = mean(corrAns, na.rm = TRUE),  
 sd = sd(corrAns, na.rm = TRUE),  
 se = sd / sqrt(N)  
 )

## `summarise()` has grouped output by 'pe', 'participant', 'conf'. You can  
## override using the `.groups` argument.

PLOT

# Violin plot of hit rate as a function of pe, grouped by confidence levels  
ggplot(df\_corr\_ans\_all, aes(x = pe, y = mean\_corr\_ans, fill = pe)) +  
 geom\_violin(color = "white", trim = TRUE) +  
 geom\_jitter(aes(x = pe, y = mean\_corr\_ans, group = participant, alpha = 0.5), color = "gray50", size = 0.5, position = position\_dodge(0)) +  
 geom\_line(aes(x = pe, y = mean\_corr\_ans, group = participant, alpha = 0.5), color = "gray50", position = position\_dodge(0)) +  
 geom\_point(data = df\_corr\_ans\_conf, size = 1) +  
 geom\_errorbar(data = df\_corr\_ans\_conf, aes(ymin = mean\_corr\_ans - se, ymax = mean\_corr\_ans + se, width = 0.1)) +  
 geom\_line(data = df\_corr\_ans\_conf, aes(x = pe, y = mean\_corr\_ans), size = 0.9, group = 1) +  
 scale\_x\_discrete(name = "PE Level", labels = c("lowPE" = "Low", "medPE" = "Medium", "highPE" = "High")) +  
 scale\_fill\_manual(values = met.brewer("Derain", 5, type = "continuous")) +  
 theme\_classic() +  
 theme(  
 text = element\_text(family = "TT Times New Roman", size = 15),   
 legend.title = element\_text(size = 1),   
 legend.text = element\_text(size = 12),   
 legend.position = "None",   
 legend.direction = "vertical") +  
 ylab("Hit Responses") +  
 ylim(0, 1) +  
 # geom\_signif(comparisons = list(c("low", "dprimeMed"), c("dprimeLow", "dprimeHigh"), c("dprimeMed", "dprimeHigh")), map\_signif\_level=TRUE) +  
 facet\_grid(conf ~ study)

