

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

```
library("tidyverse")
```

```
library("ggeffects")
```

```
library("ggplot2")
```

```
Rawdata <- read_csv("Tapping-Inner speech and purchasing  
behaviour_reversed.csv")
```

```
library(dplyr)
```

```
summary(Rawdata)
```

```
# Rename and Combine columns to create a new column
```

```
Rawdata <- Rawdata%>%
```

```
  rename(`Timing_1A2B_a1_Page_Submit` = `Timing_1A2B_a1_Page Submit`,  
  `Timing_1A2B_a2_Page_Submit` = `Timing_1A2B_a2_Page Submit`,  
  `Timing_1A2B_a3_Page_Submit` = `Timing_1A2B_a3_Page Submit`)
```

```
Rawdata <- Rawdata%>%
```

```
  rename(`Timing_1A2B_b1_Page_Submit` = `Timing_1A2B_b1_Page Submit`,  
  `Timing_1A2B_b2_Page_Submit` = `Timing_1A2B_b2_Page Submit`,  
  `Timing_1A2B_b3_Page_Submit` = `Timing_1A2B_b3_Page Submit`)
```

```
Rawdata <- Rawdata%>%
```

```
  rename(`Timing_1B2A_a1_Page_Submit` = `Timing_1B2A_a1_Page Submit`,  
  `Timing_1B2A_a2_Page_Submit` = `Timing_1B2A_a2_Page Submit`,  
  `Timing_1B2A_a3_Page_Submit` = `Timing_1B2A_a3_Page Submit`)
```

```
Rawdata <- Rawdata%>%
```

```
  rename(`Timing_1B2A_b1_Page_Submit` = `Timing_1B2A_b1_Page Submit`,  
  `Timing_1B2A_b2_Page_Submit` = `Timing_1B2A_b2_Page Submit`,  
  `Timing_1B2A_b3_Page_Submit` = `Timing_1B2A_b3_Page Submit`)
```

```
Rawdata <- Rawdata%>%
```

```
  rename(`Timing_2A1B_a1_Page_Submit` = `Timing_2A1B_a1_Page Submit`,  
  `Timing_2A1B_a2_Page_Submit` = `Timing_2A1B_a2_Page Submit`,  
  `Timing_2A1B_a3_Page_Submit` = `Timing_2A1B_a3_Page Submit`)
```

```
Rawdata <- Rawdata%>%
```

```
  rename(`Timing_2A1B_b1_Page_Submit` = `Timing_2A1B_b1_Page Submit`,  
  `Timing_2A1B_b2_Page_Submit` = `Timing_2A1B_b2_Page Submit`,  
  `Timing_2A1B_b3_Page_Submit` = `Timing_2A1B_b3_Page Submit`)
```

```
Rawdata <- Rawdata%>%
```

```

    rename( `Timing_2B1A_a1_Page_Submit` = `Timing_2B1A_a1_Page Submit`,
`Timing_2B1A_a2_Page_Submit` = `Timing_2B1A_a2_Page Submit`,
`Timing_2B1A_a3_Page_Submit` = `Timing_2B1A_a3_Page Submit`)
Rawdata <- Rawdata%>%

    rename( `Timing_2B1A_b1_Page_Submit` = `Timing_2B1A_b1_Page Submit`,
`Timing_2B1A_b2_Page_Submit` = `Timing_2B1A_b2_Page Submit`,
`Timing_2B1A_b3_Page_Submit` = `Timing_2B1A_b3_Page Submit`)

Rawdata <- Rawdata%>%

    mutate( G1Suppression_average = (Timing_1A2B_a1_Page_Submit +
Timing_1A2B_a2_Page_Submit +Timing_1A2B_a3_Page_Submit) / 3)
Rawdata <- Rawdata%>%

    mutate( G1Tapping_average = (Timing_1A2B_b1_Page_Submit +
Timing_1A2B_b2_Page_Submit + Timing_1A2B_b3_Page_Submit) / 3)
Rawdata <- Rawdata%>%

    mutate( G2Suppression_average = (Timing_1B2A_b1_Page_Submit +
Timing_1B2A_b2_Page_Submit + Timing_1B2A_b3_Page_Submit) / 3)
Rawdata <- Rawdata%>%

    mutate( G2Tapping_average = (Timing_1B2A_a1_Page_Submit +
Timing_1B2A_a2_Page_Submit + Timing_1B2A_a3_Page_Submit) / 3)
Rawdata <- Rawdata%>%

    mutate( G3Suppression_average = (Timing_2A1B_b1_Page_Submit +
Timing_2A1B_b2_Page_Submit + Timing_2A1B_b3_Page_Submit) / 3)
Rawdata <- Rawdata%>%

    mutate( G3Tapping_average = (Timing_2A1B_a1_Page_Submit +
Timing_2A1B_a2_Page_Submit + Timing_2A1B_a3_Page_Submit) / 3)
Rawdata <- Rawdata%>%

    mutate( G4Suppression_average = (Timing_2B1A_a1_Page_Submit +
Timing_2B1A_a2_Page_Submit + Timing_2B1A_a3_Page_Submit) / 3)
Rawdata <- Rawdata%>%

    mutate( G4Tapping_average = (Timing_2B1A_b1_Page_Submit +
Timing_2B1A_b2_Page_Submit + Timing_2B1A_b3_Page_Submit) / 3)

Rawdata <- Rawdata%>%

    rename( `inner_speech_QS_1` = `inner speech QS_1`, `inner_speech_QS_2` =
`inner speech QS_2`, `inner_speech_QS_3` = `inner speech QS_3`,
`inner_speech_QS_4` = `inner speech QS_4`, `inner_speech_QS_5` =

```

```
`inner_speech_QS_5`, `inner_speech_QS_6` = `inner_speech_QS_6`,
      `inner_speech_QS_7` = `inner_speech_QS_7`, `inner_speech_QS_8` =
`inner_speech_QS_8`, `inner_speech_QS_9` = `inner_speech_QS_9`,
      `inner_speech_QS_10` = `inner_speech_QS_10`, `inner_speech_QS_11` =
`inner_speech_QS_11`, `inner_speech_QS_12` = `inner_speech_QS_12`,
      `inner_speech_QS_13` = `inner_speech_QS_13`, `inner_speech_QS_14` =
`inner_speech_QS_14`, `inner_speech_QS_15` = `inner_speech_QS_15`,
      `inner_speech_QS_16` = `inner_speech_QS_16`, `inner_speech_QS_17` =
`inner_speech_QS_17`, `inner_speech_QS_18` = `inner_speech_QS_18`,
      `inner_speech_QS_19` = `inner_speech_QS_19`, `inner_speech_QS_20` =
`inner_speech_QS_20`)
```

```
library(dplyr)
```

```
Rawdata <- Rawdata %>%
```

```
  mutate(Participants_number = 1:30)
```

```
Rawdata <- Rawdata %>%
```

```
  mutate(Innerspeech_total = inner_speech_QS_1 + inner_speech_QS_2
+ inner_speech_QS_3 + inner_speech_QS_4 + inner_speech_QS_5 +
      inner_speech_QS_6 +
inner_speech_QS_7 + inner_speech_QS_8 + inner_speech_QS_10 + inner_speech_QS_1
1 + inner_speech_QS_12 +
```

```
inner_speech_QS_13 + inner_speech_QS_14 + inner_speech_QS_15 + inner_speech_QS
_16 + inner_speech_QS_17 + inner_speech_QS_18 + inner_speech_QS_19 + inner_speec
h_QS_20)
```

```
Rawdata <- Rawdata %>%
```

```
  mutate(IS_Dialogic = (inner_speech_QS_2 + inner_speech_QS_6 +
inner_speech_QS_10 + inner_speech_QS_13)/4)
```

```
Rawdata <- Rawdata %>%
```

```
  mutate(IS_Evaluate.critical = (inner_speech_QS_9 + inner_speech_QS_11 +
inner_speech_QS_17 + inner_speech_QS_18 + inner_speech_QS_20)/5)
```

```
Rawdata <- Rawdata %>%
```

```
  mutate(IS_Otherpeople = (inner_speech_QS_3 + inner_speech_QS_4 +
inner_speech_QS_5 + inner_speech_QS_12 + inner_speech_QS_16)/5)
```

```
Rawdata <- Rawdata %>%
```

```
  mutate(IS_Condensed = (inner_speech_QS_1 + inner_speech_QS_7 +
inner_speech_QS_8 + inner_speech_QS_14 + inner_speech_QS_15)/5)
```

```
Rawdata <- Rawdata %>%
```

```
mutate(IS_Regulate = inner_speech_QS_19)
```

```
Rawdata$perfume1_search_s <-  
coalesce(Rawdata$Rating_1A2B_a1_1,Rawdata$Rating_2B1A_a1_1)  
Rawdata$perfume1_search_c <-  
coalesce(Rawdata$Rating_2A1B_a1_1,Rawdata$Rating_1B2A_a1_1)
```

```
Rawdata$perfume2_search_s<-  
coalesce(Rawdata$Rating_2A1B_b1_1,Rawdata$Rating_1B2A_b1_1)  
Rawdata$perfume2_search_c<-coalesce(Rawdata$Rating_1A2B_b1_1,  
Rawdata$Rating_2B1A_b1_1)
```

```
Rawdata$bluetooth1_search_s <-  
coalesce(Rawdata$Rating_1A2B_a2_1,Rawdata$Rating_2B1A_a2_1)  
Rawdata$bluetooth1_search_c <-  
coalesce(Rawdata$Rating_2A1B_a2_1,Rawdata$Rating_1B2A_a2_1)
```

```
Rawdata$bluetooth2_search_s <-  
coalesce(Rawdata$Rating_2A1B_b2_1,Rawdata$Rating_1B2A_b2_1)  
Rawdata$bluetooth2_search_c <-  
coalesce(Rawdata$Rating_1A2B_b2_1,Rawdata$Rating_2B1A_b2_1)
```

```
Rawdata$mask1_search_s <-  
coalesce(Rawdata$Rating_1A2B_a3_1,Rawdata$Rating_2B1A_a3_1)  
Rawdata$mask1_search_c <-coalesce(Rawdata$Rating_2A1B_a3_1,  
Rawdata$Rating_1B2A_a3_1)
```

```
Rawdata$mask2_search_s <-  
coalesce(Rawdata$Rating_2A1B_b3_1,Rawdata$Rating_1B2A_b3_1)  
Rawdata$mask2_search_c<-coalesce(Rawdata$Rating_1A2B_b3_1,  
Rawdata$Rating_2B1A_b3_1)
```

```
Rawdata$perfume1_eva_s <-  
coalesce(Rawdata$Rating_1A2B_a1_2,Rawdata$Rating_2B1A_a1_2)  
Rawdata$perfume1_eva_c <-
```

```
coalesce(Rawdata$Rating_2A1B_a1_2,Rawdata$Rating_1B2A_a1_2)
```

```
Rawdata$perfume2_eva_s<-
```

```
coalesce(Rawdata$Rating_2A1B_b1_2,Rawdata$Rating_1B2A_b1_2)
```

```
Rawdata$perfume2_eva_c<-coalesce(Rawdata$Rating_1A2B_b1_2,  
Rawdata$Rating_2B1A_b1_2)
```

```
Rawdata$bluetooth1_eva_s <-
```

```
coalesce(Rawdata$Rating_1A2B_a2_2,Rawdata$Rating_2B1A_a2_2)
```

```
Rawdata$bluetooth1_eva_c <-
```

```
coalesce(Rawdata$Rating_2A1B_a2_2,Rawdata$Rating_1B2A_a2_2)
```

```
Rawdata$bluetooth2_eva_s <-
```

```
coalesce(Rawdata$Rating_2A1B_b2_2,Rawdata$Rating_1B2A_b2_2)
```

```
Rawdata$bluetooth2_eva_c <-
```

```
coalesce(Rawdata$Rating_1A2B_b2_2,Rawdata$Rating_2B1A_b2_2)
```

```
Rawdata$mask1_eva_s <-
```

```
coalesce(Rawdata$Rating_1A2B_a3_2,Rawdata$Rating_2B1A_a3_2)
```

```
Rawdata$mask1_eva_c <-coalesce(Rawdata$Rating_2A1B_a3_2,  
Rawdata$Rating_1B2A_a3_2)
```

```
Rawdata$mask2_eva_s <-
```

```
coalesce(Rawdata$Rating_2A1B_b3_2,Rawdata$Rating_1B2A_b3_2)
```

```
Rawdata$mask2_eva_c<-coalesce(Rawdata$Rating_1A2B_b3_2,  
Rawdata$Rating_2B1A_b3_2)
```

```
data_analyse_long2 <- pivot_longer(Rawdata,  
                                     cols =
```

```
c(perfume1_search_s,perfume1_search_c,
```

```
perfume2_search_s,perfume2_search_c,
```

```
bluetooth1_search_s,
```

```
bluetooth1_search_c,
```

```
bluetooth2_search_s,
```

```
bluetooth2_search_c,
```

```
mask1_search_s,
```

```
mask1_search_c,
```

```

mask2_search_s,
mask2_search_c),
names_to = "Product_sets", values_to =
"search_value")
data_analyse_cleaned2 <-
data_analyse_long2[complete.cases(data_analyse_long2$search_value), ]
data_for_search <-select(data_analyse_cleaned2, Participants_number,
                        Product_sets, search_value)
data_for_search$Independent_Condition <- NA
data_for_search$Independent_Condition[data_for_search$Product_sets %in% c(
  "perfume1_search_s", "perfume2_search_s", "bluetooth1_search_s",
  "bluetooth2_search_s", "mask1_search_s", "mask2_search_s")] <- "0.5"
data_for_search$Independent_Condition[data_for_search$Product_sets %in% c(
  "perfume1_search_c", "perfume2_search_c", "bluetooth1_search_c",
  "bluetooth2_search_c", "mask1_search_c", "mask2_search_c")] <- "-0.5"
library(tidyverse)

data_analyse_long3 <- pivot_longer(Rawdata,
                                   cols =
c(perfume1_eva_s,perfume1_eva_c,

perfume2_eva_s,perfume2_eva_c,

                                   bluetooth1_eva_s,
bluetooth1_eva_c,
                                   bluetooth2_eva_s,
bluetooth2_eva_c,
                                   mask1_eva_s, mask1_eva_c,
                                   mask2_eva_s, mask2_eva_c),
                                   names_to = "Product_sets", values_to =
"eva_value")
data_analyse_cleaned3 <-
data_analyse_long3[complete.cases(data_analyse_long3$eva_value), ]
data_for_eva <-select(data_analyse_cleaned3, Participants_number,
                    Product_sets, eva_value)
data_for_eva$Independent_Condition <- NA
data_for_eva$Independent_Condition[data_for_eva$Product_sets %in% c(
  "perfume1_eva_s", "perfume2_eva_s", "bluetooth1_eva_s",
  "bluetooth2_eva_s", "mask1_eva_s", "mask2_eva_s")] <- "0.5"

```

```
data_for_eva$Independent_Condition[data_for_eva$Product_sets %in% c(
  "perfume1_eva_c", "perfume2_eva_c", "bluetooth1_eva_c",
  "bluetooth2_eva_c", "mask1_eva_c", "mask2_eva_c")] <- "-0.5"
```

```
Rawdata$Suppression_average <- coalesce(Rawdata$G1Suppression_average,
Rawdata$G2Suppression_average, Rawdata$G3Suppression_average,
Rawdata$G4Suppression_average)
```

```
Rawdata$Tapping_average <- coalesce(Rawdata$G1Tapping_average,
Rawdata$G2Tapping_average, Rawdata$G3Tapping_average,
Rawdata$G4Tapping_average)
```

```
Rawdata$S_Infosearch_average <- coalesce((Rawdata$Rating_1A2B_a1_1 +
Rawdata$Rating_1A2B_a2_1 + Rawdata$Rating_1A2B_a3_1) / 3,
(Rawdata$Rating_2A1B_b1_1 +
Rawdata$Rating_2A1B_b2_1 + Rawdata$Rating_2A1B_b3_1) / 3,
(Rawdata$Rating_1B2A_b1_1 +
Rawdata$Rating_1B2A_b2_1 + Rawdata$Rating_1B2A_b3_1) / 3,
(Rawdata$Rating_2B1A_a1_1 +
Rawdata$Rating_2B1A_a2_1 + Rawdata$Rating_2B1A_a3_1) / 3)
```

```
Rawdata$T_Infosearch_average <- coalesce((Rawdata$Rating_1A2B_b1_1 +
Rawdata$Rating_1A2B_b2_1 + Rawdata$Rating_1A2B_b3_1) / 3,
(Rawdata$Rating_2A1B_a1_1 +
Rawdata$Rating_2A1B_a2_1 + Rawdata$Rating_2A1B_a3_1) / 3,
(Rawdata$Rating_1B2A_a1_1 +
Rawdata$Rating_1B2A_a2_1 + Rawdata$Rating_1B2A_a3_1) / 3,
(Rawdata$Rating_2B1A_b1_1 +
Rawdata$Rating_2B1A_b2_1 + Rawdata$Rating_2B1A_b3_1) / 3,)
```

```
Rawdata$S_Altevaluation_average <- coalesce((Rawdata$Rating_1A2B_a1_2 +
Rawdata$Rating_1A2B_a2_2 + Rawdata$Rating_1A2B_a3_2) / 3,
(Rawdata$Rating_2A1B_b1_2 +
+ Rawdata$Rating_2A1B_b2_2 + Rawdata$Rating_2A1B_b3_2) / 3,
(Rawdata$Rating_1B2A_b1_2 +
+ Rawdata$Rating_1B2A_b2_2 + Rawdata$Rating_1B2A_b3_2) / 3,
```

```

(Rawdata$Rating_2B1A_a1_2
+ Rawdata$Rating_2B1A_a2_2 + Rawdata$Rating_2B1A_a3_2) / 3)

```

```

Rawdata$T_Altevaluation_average <- coalesce((Rawdata$Rating_1A2B_b1_2 +
Rawdata$Rating_1A2B_b2_2 + Rawdata$Rating_1A2B_b3_2) / 3,
(Rawdata$Rating_2A1B_a1_2
+ Rawdata$Rating_2A1B_a2_2 + Rawdata$Rating_2A1B_a3_2) / 3,
(Rawdata$Rating_1B2A_a1_2
+ Rawdata$Rating_1B2A_a2_2 + Rawdata$Rating_1B2A_a3_2) / 3,
(Rawdata$Rating_2B1A_b1_2
+ Rawdata$Rating_2B1A_b2_2 + Rawdata$Rating_2B1A_b3_2) / 3,)

```

```

Rawdata$Infosearch_Perfume_average <- rowSums(Rawdata[,
c("Rating_1A2B_a1_1", "Rating_1A2B_b1_1",

```

```

"Rating_1B2A_a1_1", "Rating_1B2A_b1_1",

```

```

"Rating_2A1B_a1_1", "Rating_2A1B_b1_1",

```

```

"Rating_2B1A_a1_1", "Rating_2B1A_b1_1"]], na.rm = TRUE)

```

```

Rawdata$Altevaluation_Perfume_average <- rowSums(Rawdata[,
c("Rating_1A2B_a1_2", "Rating_1A2B_b1_2",

```

```

"Rating_1B2A_a1_2", "Rating_1B2A_b1_2",

```

```

"Rating_2A1B_a1_2", "Rating_2A1B_b1_2",

```

```

"Rating_2B1A_a1_2", "Rating_2B1A_b1_2"]], na.rm = TRUE)

```

```

Rawdata$Infosearch_bluetooth_average <- rowSums(Rawdata[,
c("Rating_1A2B_a2_1", "Rating_1A2B_b2_1",

```

```

"Rating_1B2A_a2_1", "Rating_1B2A_b2_1",

```

```

"Rating_2A1B_a2_1", "Rating_2A1B_b2_1",

```

```

"Rating_2B1A_a2_1", "Rating_2B1A_b2_1"]], na.rm = TRUE)

```





```

Timing_1A2B_b2_Page_Submit, Timing_1A2B_b3_Page_Submit,
                                Timing_1B2A_a1_Page_Submit,
Timing_1B2A_a2_Page_Submit, Timing_1B2A_a3_Page_Submit,
                                Timing_1B2A_b1_Page_Submit,
Timing_1B2A_b2_Page_Submit, Timing_1B2A_b3_Page_Submit,
                                Timing_2A1B_a1_Page_Submit,
Timing_2A1B_a2_Page_Submit, Timing_2A1B_a3_Page_Submit,
                                Timing_2A1B_b1_Page_Submit,
Timing_2A1B_b2_Page_Submit, Timing_2A1B_b3_Page_Submit,
                                Timing_2B1A_a1_Page_Submit,
Timing_2B1A_a2_Page_Submit, Timing_2B1A_a3_Page_Submit,
                                Timing_2B1A_b1_Page_Submit,
Timing_2B1A_b2_Page_Submit, Timing_2B1A_b3_Page_Submit),
                                names_to = "Product_sets", values_to =
"Decision_time")

```

```

# Assuming 'data_analyse_long' is the data frame

```

```

# Remove rows with NA in "Decision_time" column

```

```

data_analyse_cleaned <-

```

```

data_analyse_long[complete.cases(data_analyse_long$Decision_time), ]

```

```

#Quantitative data which need to be analysed

```

```

data_for_gamma2 <- select(data_analyse_cleaned, Participants_number,

```

```

                                IS_Dialogic, IS_Evaluate.critical, IS_Otherpeople,

```

```

IS_Condensed, IS_Regulate, Innerspeech_total,

```

```

                                Suppression_average, Tapping_average,

```

```

                                S_Infosearch_average, T_Infosearch_average,

```

```

                                S_Altevaluation_average, T_Altevaluation_average,

```

```

Product_sets, Decision_time)

```

```

# Sample data frame with the column names

```

```

library(dplyr)

```

```

# Create a new column "Independent_Condition" and set all values to NA

```

```

data_for_gamma2$Independent_Condition <- NA

```

```

# Assign values based on conditions (0.5=suppression, -0.5=Tapping)

```

```

data_for_gamma2$Independent_Condition[data_for_gamma2$Product_sets %in% c(

```

```

    "Timing_1A2B_a1_Page_Submit", "Timing_1A2B_a2_Page_Submit",
    "Timing_1A2B_a3_Page_Submit",
    "Timing_1B2A_b1_Page_Submit", "Timing_1B2A_b2_Page_Submit",
    "Timing_1B2A_b3_Page_Submit",
    "Timing_2A1B_b1_Page_Submit", "Timing_2A1B_b2_Page_Submit",
    "Timing_2A1B_b3_Page_Submit",
    "Timing_2B1A_a1_Page_Submit", "Timing_2B1A_a2_Page_Submit",
    "Timing_2B1A_a3_Page_Submit"
  ]) <- "0.5"

```

```

data_for_gamma2$Independent_Condition[data_for_gamma2$Product_sets %in% c(
  "Timing_1A2B_b1_Page_Submit", "Timing_1A2B_b2_Page_Submit",
  "Timing_1A2B_b3_Page_Submit",
  "Timing_1B2A_a1_Page_Submit", "Timing_1B2A_a2_Page_Submit",
  "Timing_1B2A_a3_Page_Submit",
  "Timing_2A1B_a1_Page_Submit", "Timing_2A1B_a2_Page_Submit",
  "Timing_2A1B_a3_Page_Submit",
  "Timing_2B1A_b1_Page_Submit", "Timing_2B1A_b2_Page_Submit",
  "Timing_2B1A_b3_Page_Submit"
)] <- "-0.5"

```

```

data_for_gamma2 <- data_for_gamma2 %>%
  mutate(Product_sets = case_when(
    Product_sets == "Timing_1A2B_a1_Page_Submit" ~ "perfume1",
    Product_sets == "Timing_1A2B_a2_Page_Submit" ~ "bluetooth1",
    Product_sets == "Timing_1A2B_a3_Page_Submit" ~ "mask1",
    Product_sets == "Timing_1A2B_b1_Page_Submit" ~ "perfume2",
    Product_sets == "Timing_1A2B_b2_Page_Submit" ~ "bluetooth2",
    Product_sets == "Timing_1A2B_b3_Page_Submit" ~ "mask2",
    Product_sets == "Timing_1B2A_a1_Page_Submit" ~ "perfume1",
    Product_sets == "Timing_1B2A_a2_Page_Submit" ~ "bluetooth1",
    Product_sets == "Timing_1B2A_a3_Page_Submit" ~ "mask1",
    Product_sets == "Timing_1B2A_b1_Page_Submit" ~ "perfume2",
    Product_sets == "Timing_1B2A_b2_Page_Submit" ~ "bluetooth2",
    Product_sets == "Timing_1B2A_b3_Page_Submit" ~ "mask2",
    Product_sets == "Timing_2A1B_a1_Page_Submit" ~ "perfume1",
    Product_sets == "Timing_2A1B_a2_Page_Submit" ~ "bluetooth1",
    Product_sets == "Timing_2A1B_a3_Page_Submit" ~ "mask1",
  ))

```

```

Product_sets == "Timing_2A1B_b1_Page_Submit" ~ "perfume2",
Product_sets == "Timing_2A1B_b2_Page_Submit" ~ "bluetooth2",
Product_sets == "Timing_2A1B_b3_Page_Submit" ~ "mask2",
Product_sets == "Timing_2B1A_a1_Page_Submit" ~ "perfume1",
Product_sets == "Timing_2B1A_a2_Page_Submit" ~ "bluetooth1",
Product_sets == "Timing_2B1A_a3_Page_Submit" ~ "mask1",
Product_sets == "Timing_2B1A_b1_Page_Submit" ~ "perfume2",
Product_sets == "Timing_2B1A_b2_Page_Submit" ~ "bluetooth2",
Product_sets == "Timing_2B1A_b3_Page_Submit" ~ "mask2",
TRUE ~ NA_character_
))

```

```

data_analyse$Suppression_average <-
as.numeric(data_analyse$Suppression_average)
data_analyse$Tapping_average <- as.numeric(data_analyse$Tapping_average)

```

```

#t-test 1. Decision Timing./ 2.Information Search rating score / 3. Alternatives
Evaluation rating score
t.test(data_analyse$Tapping_average, data_analyse$Suppression_average, paired =
TRUE)
t.test(data_analyse$T_Infsearch_average, data_analyse$S_Infsearch_average,
paired = TRUE)
t.test(data_analyse$T_Altevaluation_average,
data_analyse$S_Altevaluation_average, paired = TRUE)

```

##### Effect size

### 1 Average time

# mean difference between two groups

mean\_difference\_at <- 7.580722

# Sstandard deviation of the data

```

sd_averagetime <- sd(c(data_analyse$Tapping_average,
data_analyse$Suppression_average)) # Replace group1 and group2 with your
actual data

```

```

# Calculate Cohen's d
cohens_d_average_time <- mean_difference_at / sd_averagetime

# Print the effect size
print(cohens_d_average_time)

### 2 Information search
# mean difference between two groups
mean_difference_is <- 0.9444444
# Standard deviation of the data
sd_info <- sd(c(data_analyse$T_Infosearch_average,
data_analyse$S_Infosearch_average)) # Replace group1 and group2 with your
actual data

# Calculate Cohen's d
cohens_d_infosearch <- mean_difference_is / sd_info
# Print the effect size
print(cohens_d_infosearch)

### 3 Alternatives Evaluation
# mean difference between two groups
mean_difference_ae <- 0.5555556
# Standard deviation of the data
sd_alt <- sd(c(data_analyse$T_Altevaluation_average,
data_analyse$S_Altevaluation_average)) # Replace group1 and group2 with your
actual data

# Calculate Cohen's d
cohens_d_altevaluation <- mean_difference_ae / sd_alt
# Print the effect size
print(cohens_d_altevaluation)

# Average Decision time
library(ggplot2)

# Create a new dataframe with relevant columns
Averagetime <- data.frame(

```

```

Value = c(data_analyse$Suppression_average, data_analyse$Tapping_average),
Condition = rep(c("Suppression Decision Time", "Tapping Decision Time"), each =
nrow(data_analyse)),
Participants_number = rep(data_analyse$Participants_number, 2))
Averagetime$Condition <- factor(Averagetime$Condition, levels = c("Tapping
Decision Time", "Suppression Decision Time"))

```

```

# Create the violin plot with connected dots and overlay boxplots
ggplot(Averagetime, aes(x = Condition, y = Value, fill = Condition)) +
  geom_violin(trim = FALSE, alpha = 0.4) + # Make the violins more transparent
  geom_point(position = "identity", width = 0.2, alpha = 0.3) + # Make the points
more transparent
  geom_boxplot(width = 0.2, alpha = 0.8) + # Overlay boxplots on top
  geom_line(aes(group = Participants_number), color = "black", alpha = 0.3) + #
Make the lines more transparent
  labs(title = "Comparison of Suppression and Tapping on Average Decision Time",
x = "Condition", y = "Decision time") +
  theme_minimal()

```

# Information search score

#Violin plots

# Create a new dataframe with relevant columns

```

Infosearch <- data.frame(
  Value = c(data_analyse$S_Infosearch_average,
data_analyse$T_Infosearch_average),
  Condition = rep(c("Suppression Information Search Score", "Tapping Information
Search Score"), each = nrow(data_analyse)),
  Participants_number = rep(data_analyse$Participants_number, 2))
Infosearch$Condition <- factor(Infosearch$Condition, levels = c("Tapping Information
Search Score", "Suppression Information Search Score"))

```

# Create the violin plot with connected dots and box plot

```

ggplot(Infosearch, aes(x = Condition, y = Value, fill = Condition)) +
  geom_violin(trim = FALSE, alpha = 0.4) +
  geom_point(position = "identity", width = 0.2) +
  geom_boxplot(width = 0.2, alpha = 0.8) +
  geom_line(aes(group = Participants_number), color = "black", alpha = 0.3) +
  labs(title = "Comparison of Information Search rate in two conditions",

```

```
x = "Condition", y = "Rate Score") +  
theme_minimal()
```

```
#Alternatives evaluation score
```

```
#violin plots for alternative evaluation
```

```
# Create a new dataframe with relevant columns
```

```
Altevaluation <- data.frame(  
  Value = c(data_analyse$S_Altevaluation_average,  
data_analyse$T_Altevaluation_average),  
  Condition = rep(c("Suppression Alternative Evaluation Score", "Tapping Alternative  
Evaluation Score"), each = nrow(data_analyse)),  
  Participants_number = rep(data_analyse$Participants_number, 2))  
Altevaluation$Condition <- factor(Altevaluation$Condition, levels = c("Tapping  
Alternative Evaluation Score", "Suppression Alternative Evaluation Score"))
```

```
# Create the violin plot with connected dots and box plot
```

```
ggplot(Altevaluation, aes(x = Condition, y = Value, fill = Condition)) +  
  geom_violin(trim = FALSE, alpha = 0.4) +  
  geom_point(position = "identity", width = 0.2) +  
  geom_boxplot(width = 0.2, alpha = 0.8) +  
  geom_line(aes(group = Participants_number), color = "black", alpha = 0.3) +  
  labs(title = "Comparison of Alternatives Evaluation rate in two conditions",  
        x = "Condition", y = "Rate Score") +  
  theme_minimal()
```

```
library(ggplot2)
```

```
library(patchwork)
```

```
# Your individual ggplot objects
```

```
plot1 <- ggplot(Averagetime, aes(x = Condition, y = Value, fill = Condition)) +  
  geom_violin(trim = FALSE, alpha = 0.4) +  
  geom_point(position = "identity", width = 0.2, alpha = 0.3) +  
  geom_boxplot(width = 0.2, alpha = 0.8) +  
  geom_line(aes(group = Participants_number), color = "black", alpha = 0.3) +
```

```

labs(title = "Comparison of Suppression and Tapping on Average Decision Time",
      x = "Condition", y = "Decision time") +
theme_minimal()

plot2 <- ggplot(Infosearch, aes(x = Condition, y = Value, fill = Condition)) +
  geom_violin(trim = FALSE, alpha = 0.4) +
  geom_point(position = "identity", width = 0.2) +
  geom_boxplot(width = 0.2, alpha = 0.8) +
  geom_line(aes(group = Participants_number), color = "black", alpha = 0.3) +
  labs(title = "Comparison of Information Search rate in two conditions",
        x = "Condition", y = "Rate Score") +
  theme_minimal()

plot3 <- ggplot(Altevaluation, aes(x = Condition, y = Value, fill = Condition)) +
  geom_violin(trim = FALSE, alpha = 0.4) +
  geom_point(position = "identity", width = 0.2) +
  geom_boxplot(width = 0.2, alpha = 0.8) +
  geom_line(aes(group = Participants_number), color = "black", alpha = 0.3) +
  labs(title = "Comparison of Alternatives Evaluation rate in two conditions",
        x = "Condition", y = "Rate Score") +
  theme_minimal()

#Linear model for inner speech frequency and decision time
library(lme4)

data_for_gamma2$Participants_number <-
as.numeric(data_for_gamma2$Participants_number)
data_for_gamma2$Product_sets <- as.factor(data_for_gamma2$Product_sets)
data_for_gamma2$Independent_Condition <-
ifelse(data_for_gamma2$Independent_Condition == "0.5", 0.5, -0.5)
data_for_gamma2 = data_for_gamma2 %>%
  mutate(IS_D = scale(IS_Dialogic)/2,
         IS_C = scale(IS_Condensed)/2,
         IS_E = scale(IS_Evaluate.critical)/2,
         IS_O = scale(IS_Otherpeople)/2,
         IS_R = scale(IS_Regulate)/2)

quantiles <- quantile(data_for_gamma2$Decision_time, probs = c(0.005, 0.995))

```



```

loend <- quantiles[1]
hiend <- quantiles[2]

# Initial trimming
data_for_gamma2 <- data_for_gamma2 %>%
  filter(Decision_time > loend & Decision_time < hiend)

# Grouping and secondary trimming
data_for_gamma2 <- data_for_gamma2 %>%
  group_by(Independent_Condition, IS_D, IS_C, IS_E, IS_R, IS_O) %>%
  filter(Decision_time <= mean(Decision_time) + 3 * sd(Decision_time) &
    Decision_time >= mean(Decision_time) - 3 * sd(Decision_time))

model <- glmer(Decision_time ~ Independent_Condition * IS_D +
  Independent_Condition * IS_C +
    Independent_Condition * IS_E + Independent_Condition *
  IS_O +
    Independent_Condition * IS_R +
    (0 +Independent_Condition | Participants_number) +
    (1 +Independent_Condition | Product_sets), data =
  data_for_gamma2, family = Gamma(link = "identity"),
  glmerControl(optimizer = "bobyqa"))
summary(model)

```

```

### decision time
quantiles <- quantile(data_for_gamma2$Decision_time, probs = c(0.010, 0.990))
loend <- quantiles[1]
hiend <- quantiles[2]

```

```

# Initial trimming
data_for_gamma2 <- data_for_gamma2 %>%
  filter(Decision_time > loend & Decision_time < hiend)

# Grouping and secondary trimming
data_for_gamma2 <- data_for_gamma2 %>%
  group_by(Independent_Condition) %>%
  filter(Decision_time <= mean(Decision_time) + 3 * sd(Decision_time) &

```

```
Decision_time >= mean(Decision_time) - 3 * sd(Decision_time))
```

```
##### Have to trim the data to 173 obs with the same number of participants
```

```
model2 <- glmer(  
  Decision_time ~ Independent_Condition +  
    (1 + Independent_Condition | Participants_number) + (1 +  
Independent_Condition | Product_sets),  
  data = data_for_gamma2, family = Gamma(link = "identity"),  
  control = glmerControl(optimizer = "bobyqa"))
```

```
summary(model2)
```

```
fixed_effects <- fixef(model2)  
print(fixed_effects)
```

```
library(stringr)
```

```
### info search model
```

```
data_for_search <- data_for_search %>%  
  mutate(Product_sets = case_when(  
    str_detect(Product_sets, "perfume1") ~ "perfume1",  
    str_detect(Product_sets, "bluetooth1") ~ "bluetooth1",  
    str_detect(Product_sets, "mask1") ~ "mask1",  
    str_detect(Product_sets, "perfume2") ~ "perfume2",  
    str_detect(Product_sets, "bluetooth2") ~ "bluetooth2",  
    str_detect(Product_sets, "mask2") ~ "mask2",  
    TRUE ~ Product_sets  
  ))
```

```
library(ordinal)
```

```
data_for_search$search_value <- factor(data_for_search$search_value, ordered =  
TRUE)
```

```
model3 <- clmm(search_value ~ Independent_Condition + (1 +  
Independent_Condition | Participants_number) + (1 + Independent_Condition |  
Product_sets),
```

```
  data = data_for_search)
```

```
summary(model3)
```

```
### alt eva model
```

```
data_for_eva <- data_for_eva %>%
```

```
  mutate(Product_sets = case_when(  
    str_detect(Product_sets, "perfume1") ~ "perfume1",  
    str_detect(Product_sets, "bluetooth1") ~ "bluetooth1",  
    str_detect(Product_sets, "mask1") ~ "mask1",  
    str_detect(Product_sets, "perfume2") ~ "perfume2",  
    str_detect(Product_sets, "bluetooth2") ~ "bluetooth2",  
    str_detect(Product_sets, "mask2") ~ "mask2",  
    TRUE ~ Product_sets  
  ))
```

```
data_for_eva$eva_value <- factor(data_for_eva$eva_value, ordered = TRUE)
```

```
model4 <- clmm(eva_value ~ Independent_Condition + (1 + Independent_Condition |  
Participants_number) + (1 + Independent_Condition | Product_sets),  
              data = data_for_eva)  
summary(model4)
```

```
library(car)
```

```
vif(model)
```

```
summary(data_for_gamma2)  
sd(data_for_gamma2$IS_Dialogic)  
sd(data_for_gamma2$IS_Evaluate.critical)  
sd(data_for_gamma2$IS_Otherpeople)  
sd(data_for_gamma2$IS_Condensed)  
sd(data_for_gamma2$IS_Regulate)  
sd(data_for_gamma2$Suppression_average)  
sd(data_for_gamma2$Tapping_average)  
sd(data_for_gamma2$S_Infosearch_average)  
sd(data_for_gamma2$T_Infosearch_average)  
sd(data_for_gamma2$S_Altevaluation_average)  
sd(data_for_gamma2$T_Altevaluation_average)  
sd(data_for_gamma2$Decision_time)
```