

Análisis de resultados: Segunda aplicación

Contents

1	Introducción	5
2	Obtención de datos	7
3	Análisis de datos	8
3.1	Actitudes	8
3.1.1	Actitudes hacia el lenguaje	9
3.1.1.1	Alpha	11
3.1.1.1.1	Total pre	11
3.1.1.1.2	Total post	11
3.1.1.1.3	Componente cognitivo pre	12
3.1.1.1.4	Componente cognitivo post	12
3.1.1.1.5	Componente afectivo pre	13
3.1.1.1.6	Componente afectivo post	13
3.1.1.1.7	Componente conativo pre	13
3.1.1.1.8	Componente conativo post	14
3.1.1.2	Indicadores psicométricos	14
3.1.1.2.1	Total pre	15
3.1.1.2.2	Total post	16
3.1.1.3	Comparación pre-post	16
3.1.1.3.1	Prueba total	17
3.1.1.3.2	Componente afectivo	20
3.1.1.3.3	Componente Cognitivo	23
3.1.1.3.4	Componente Conativo	26
3.1.2	Actitudes hacia las matemáticas	29
3.1.2.1	Alpha	30
3.1.2.1.1	Total pre	30
3.1.2.1.2	Total post	31
3.1.2.1.3	Componente cognitivo pre	32
3.1.2.1.4	Componente cognitivo post	32
3.1.2.1.5	Componente afectivo pre	32
3.1.2.1.6	Componente afectivo post	33
3.1.2.1.7	Componente conativo pre	33
3.1.2.1.8	Componente conativo post	33
3.1.2.2	Indicadores psicométricos	34
3.1.2.2.1	Total pre	34
3.1.2.2.2	Total post	35
3.1.2.3	Comparación pre-post	37

	3.1.2.3.1	Prueba total	37
	3.1.2.3.2	Componente afectivo	40
	3.1.2.3.3	Componente Cognitivo	43
	3.1.2.3.4	Componente Conativo	46
3.2	Motivación		49
3.2.1	Alpha		53
	3.2.1.1	Total pre	53
	3.2.1.2	Total post	53
	3.2.1.3	Interés pre	54
	3.2.1.4	Interés post	54
	3.2.1.5	Metas pre	55
	3.2.1.6	Metas post	55
	3.2.1.7	Atribución interna pre	55
	3.2.1.8	Atribución interna pre	55
	3.2.1.9	Expectativa positiva pre	56
	3.2.1.10	Expectativa positiva post	56
3.2.2	Indicadores psicométricos		56
	3.2.2.1	Total pre	56
	3.2.2.2	Total post	57
3.2.3	Comparación pre-post		58
	3.2.3.1	Prueba total	58
	3.2.3.1.1	Estadísticos de normalidad	58
	3.2.3.1.2	Descriptivos	59
	3.2.3.1.3	Comparación de medias	59
	3.2.3.1.4	Tamaño del efecto	60
	3.2.3.2	Interés	61
	3.2.3.2.1	Estadísticos de normalidad	61
	3.2.3.2.2	Descriptivos	62
	3.2.3.2.3	Comparación de medias	62
	3.2.3.2.4	Tamaño del efecto	63
	3.2.3.3	Metas	64
	3.2.3.3.1	Estadísticos de normalidad	64
	3.2.3.3.2	Descriptivos	65
	3.2.3.3.3	Comparación de medias	65
	3.2.3.3.4	Tamaño del efecto	66
	3.2.3.4	Atribución interna	67
	3.2.3.4.1	Estadísticos de normalidad	67
	3.2.3.4.2	Descriptivos	68
	3.2.3.4.3	Comparación de medias	68
	3.2.3.4.4	Tamaño del efecto	69
	3.2.3.5	Expectativas positivas	70
	3.2.3.5.1	Estadísticos de normalidad	70
	3.2.3.5.2	Descriptivos	71
	3.2.3.5.3	Comparación de medias	71
	3.2.3.5.4	Tamaño del efecto	72
3.3	Funciones ejecutivas		73
3.3.1	Memoria auditiva		73
	3.3.1.1	Alpha	76

	3.3.1.1.1	Pre	76
	3.3.1.1.2	Post	77
3.3.1.2		Indicadores psicométricos	78
	3.3.1.2.1	Pre	78
	3.3.1.2.2	Post	80
3.3.1.3		Comparación pre-post	82
	3.3.1.3.1	Estadísticos de normalidad	82
	3.3.1.3.2	Descriptivos	83
	3.3.1.3.3	Comparación de medias	84
	3.3.1.3.4	Tamaño del efecto	85
3.3.2		Memoria visual	86
	3.3.2.1	Alpha	88
	3.3.2.1.1	Pre	88
	3.3.2.1.2	Post	89
3.3.2.2		Indicadores psicométricos	90
	3.3.2.2.1	Pre	90
	3.3.2.2.2	Post	92
3.3.2.3		Comparación pre-post	94
	3.3.2.3.1	Estadísticos de normalidad	94
	3.3.2.3.2	Descriptivos	95
	3.3.2.3.3	Comparación de medias	95
	3.3.2.3.4	Tamaño del efecto	97
3.3.3		Inhibición	97
	3.3.3.1	Alpha	99
	3.3.3.1.1	Pre	99
	3.3.3.1.2	Post	99
3.3.3.2		Indicadores psicométricos	100
	3.3.3.2.1	Pre	101
	3.3.3.2.2	Post	102
3.3.3.3		Comparación pre-post	103
	3.3.3.3.1	Estadísticos de normalidad	103
	3.3.3.3.2	Descriptivos	103
	3.3.3.3.3	Comparación de medias	104
	3.3.3.3.4	Tamaño del efecto	105
3.3.4		Flexibilidad	106
	3.3.4.1	Alpha	107
	3.3.4.1.1	Pre	108
	3.3.4.1.2	Post	108
3.3.4.2		Indicadores psicométricos	109
	3.3.4.2.1	Pre	109
	3.3.4.2.2	Post	110
3.3.4.3		Comparación pre-post	111
	3.3.4.3.1	Estadísticos de normalidad	111
	3.3.4.3.2	Descriptivos	112
	3.3.4.3.3	Comparación de medias	113
	3.3.4.3.4	Tamaño del efecto	114
3.4		Socioemocionales	115
	3.4.1	Alpha	119

3.4.1.1	Total pre	119
3.4.1.2	Total post	119
3.4.1.3	Regulación pre	120
3.4.1.4	Regulación post	120
3.4.1.5	Expresión pre	121
3.4.1.6	Expresión post	121
3.4.1.7	Reconocimiento pre	121
3.4.1.8	Reconocimiento post	122
3.4.2	Indicadores psicométricos	122
3.4.2.0.1	Total pre	123
3.4.2.0.2	Total post	124
3.4.3	Comparación pre-post	124
3.4.3.1	Regulación	125
3.4.3.1.1	Estadísticos de normalidad	125
3.4.3.1.2	Descriptivos	126
3.4.3.1.3	Comparación de medias	126
3.4.3.1.4	Tamaño del efecto	127
3.4.3.2	Reconocimiento	129
3.4.3.2.1	Estadísticos de normalidad	129
3.4.3.2.2	Descriptivos	130
3.4.3.2.3	Comparación de medias	130
3.4.3.2.4	Tamaño del efecto	131
3.4.3.3	Expresión	133
3.4.3.3.1	Estadísticos de normalidad	133
3.4.3.3.2	Descriptivos	133
3.4.3.3.3	Comparación de medias	134
3.4.3.3.4	Tamaño del efecto	135
4	Resúmenes de datos	137
4.1	Alpha de las pruebas	137
4.2	Descriptivos	138
4.3	Indicadores psicométricos	139
4.4	Pruebas pre-post	144
4.4.1	Actitudes hacia el lenguaje	146
4.4.2	Actitudes hacia las matemáticas	147
4.4.3	Motivación	148
4.4.4	Funciones ejecutivas	149
4.4.5	Habilidades socioemocionales	150
4.4.6	Diferencias entre ciudades	151
4.4.7	Diferencias entre ciclos	152
5	Información de la sesión	155

1 Introducción

Este documento tiene como objetivo reportar el análisis de resultados psicométricos y pre-post de la aplicación de pruebas de Actitudes, Funciones ejecutivas, Motivación y Habilidades socioemocionales, realizados en el marco del Proyecto de evaluación del Plan Todo al Cole desarrollado por la Fundación Pies Descalzos.

El análisis psicométrico consiste en la obtención de indicadores de calidad de los ítems y pruebas. Los indicadores utilizados se listan a continuación:

Ítems dicotómicos

- *Sample.SD* representa la desviación estándar del ítem.
- *Item.total* muestra la correlación ítem-total.
- *Item.Tot.woi* representa la correlación del ítem con el total de la prueba, excluyendo al ítem en cuestión. Este indicador está muy ligado a la confiabilidad, por lo que valores inferiores a .10 no son deseados, y valores negativos representan ítems con problemas.
- *Difficulty* la dificultad según la TCT. Para este caso, lo mejor sería que los indicadores se encontraran entre el 0.10 y el 0.90
- *Discrimination* la discriminación entre tercios. Se recomiendan valores superiores a 0.20
- *Item.Reliab* la confiabilidad del ítem. Su función es medir la contribución del ítem a la medida final del test.
- *Item.Rel.woi* la confiabilidad del ítem, excluyendo al ítem en el total del test utilizado en la fórmula. Su función es medir la contribución del ítem a la medida final del test. Este indicador es interesante a la hora de mezclar ítems de ambas formas de prueba ya que da una guía de su posible comportamiento.

Ítems en escala likert

- *Difficulty*: Dificultad desde TCT
- *Mean*: Media del ítem
- *SD*: Desviación estándar del ítem
- *Prop.max.score*: La proporción de sujetos que escogió la máxima categoría
- *RIR*: Correlación entre el ítem y el resultado de la prueba sin contar el ítem.
- *RIT*: Correlación entre el ítem y el resultado de la prueba
- *ULI*: Discriminación upper-lower
- *Alpha.drop*: Alpha de Cronbach sin el ítem
- *Index.rel*: Índice de confiabilidad del ítem

Adicionalmente, se realizó un análisis pre y post de los resultados de los estudiantes en las pruebas. Dicho análisis consistió en una comparación de medias para muestras relacionadas, mediante la prueba *W de Wilcoxon*, así también se estimó el tamaño del efecto mediante el estadístico *d de Cohen*.

```
## Librerías
```

```
# Datos
```

```

library(readxl)
library(xlsx)
library(tidyr)
library(dplyr)
library(googlesheets4)
library(stringr)

# Análisis de datos
library(likert)
library(nortest)
library(effsize)
library(psychometric)
library(ShinyItemAnalysis)

#Graficas
library(ggplot2)
library(gt)
library(DT)
library(ggtech)
library(ggthemr)
library(hrbrthemes)
library(ggthemes)

colores = c("#D94389", "#36BFB1", "#ADD96C", "#F2C230", "#F2F2F2")

#Funciones propias

source("../Functions/min_max_scaler.R")
source("../Functions/calificacion.R")
source("../Functions/change_to_zero.R")

# Otros
options(digits=5, scipen = 50)
set.seed(321)
# rmarkdown::render(input="1.0-bapinedam-segunda_aplicacion.Rmd",
#                    output_file = "../Pdf/Resultados primera aplicacion.pdf")

```

2 Obtención de datos

Para este proyecto las bases de datos se obtienen directamente desde internet, específicamente, desde google drive, debido a que pueden agregarse datos y es necesario que cada vez que se ejecute el script, los datos estén actualizados.

```
# Data
```

```
### Autenticación de usuario
```

```
gs4_auth()
```

```
# pre
```

```
### Obtención de los datos
```

```
url_pre = paste("https://docs.google.com/spreadsheets/d/",  
                "1nmTecdohkKDNcppMfIXT0yrmKnoxIg-ZQZHUI0g3LBA/edit#gid=878062551",  
                sep = "")
```

```
# post
```

```
url_post = paste("https://docs.google.com/spreadsheets/d/",  
                 "1n69qyde6BSUWVV5jSv3-sbCRF_gWyErPwatNAqKG9vo/edit#gid=878062551",  
                 sep = "")
```

3 Análisis de datos

3.1 Actitudes

En el caso de la prueba pre de actitudes, todas las claves con la B, es por ello que podemos calificar siguiendo la instrucción: Si es B entonces 1, si no, entonces 0.

```
# Pre
actitudes_pre = read_sheet(url_pre, sheet = "Actitudes")
actitudes_pre = actitudes_pre[,1:18]

col_items = paste(rep("Grupo", 12), rep(1:4, each = 3), rep(paste("_", rep(1:3, 3))))

vector = c(colnames(actitudes_pre)[1:6], col_items)
colnames(actitudes_pre) = vector

actitudes_pre = filter(actitudes_pre,
                        !is.na(`Código`),
                        !is.na(`Grupo 1 _ 1`),
                        `Grupo 1 _ 1` %in% c("A", "B", "0", "X"))
dim(actitudes_pre)

## [1] 1115 18

actitudes_pre[,7:18] = apply(actitudes_pre[,7:18], 2, function(x) str_to_upper(x))

claves_lenguaje_pre = c('B','A','A','B','B','A','A','A','B','B','A','B')
claves_matematicas_pre = c('B','A','A','B','A','A','A','B','B','A','A','B')
```

En el caso de la prueba ppost de actitudes, no todos los ítems tienen la misma clave. Es por ello que creamos una función que tome un vector con las claves y nos califique una a una las columnas.

```
actitudes_post = read_sheet(url_post, sheet = "Actitudes")
actitudes_post = actitudes_post[,1:18]

col_items = paste(rep("Grupo", 12), rep(1:4, each = 3), rep(paste("_", rep(1:3, 3))))

vector = c(colnames(actitudes_post)[1:6], col_items)
colnames(actitudes_post) = vector

actitudes_post[,7:18] = apply(actitudes_post[,7:18], 2, function(x) as.character(x))

actitudes_post = filter(actitudes_post,
                        !is.na(`Código`),
                        !is.na(`Grupo 1 _ 1`),
                        `Grupo 1 _ 1` %in% c("A", "B", "0", "X"))
```



```
actitudes_post[,7:18] = apply(actitudes_post[,7:18], 2, function(x) str_to_upper(x))
```

```
claves_lenguaje_post = c('A','A','B','B','B','B','B','A','A','A','B','A')
```

```
claves_matematicas_post = c('B','A','A','A','A','B','A','A','B','B','A','B')
```

3.1.1 Actitudes hacia el lenguaje

Todos los estudiantes tienen un código. Si el mismo empieza en 1, es porque el estudiante estuvo en el programa de lenguaje, si tiene dos, es porque estuvo en el programa de mejora de matemáticas. En este caso filtramos por el 1.

```
actitudes_lenguaje_pre = filter(actitudes_pre, substr(`Código`, 1, 1) == "1")
actitudes_lenguaje_post = filter(actitudes_post, substr(`Código`, 1, 1) == "1")
```

Calificación pre

```
actitudes_lenguaje_pre[,7:18] = calificacion(actitudes_lenguaje_pre[,7:18],
                                             claves_lenguaje_pre)
```

Calificación post

```
actitudes_lenguaje_post[,7:18] = calificacion(actitudes_lenguaje_post[,7:18],
                                              claves_lenguaje_post)
```

Finalmente, ya que tenemos calificados todos los ítems, obtenemos puntuaciones generales.

Calificación

Total

```
actitudes_lenguaje_pre$Total_pre = apply(actitudes_lenguaje_pre[,7:18], 1,
                                          function(x) sum(x, na.rm= FALSE))
```

```
actitudes_lenguaje_post$Total_post = apply(actitudes_lenguaje_post[,7:18], 1,
                                            function(x) sum(x, na.rm= FALSE))
```

Afectivo

```
actitudes_lenguaje_pre$Afectivo_pre = apply(actitudes_lenguaje_pre[,c(7, 10, 13, 16)], 1,
                                             function(x) sum(x, na.rm= FALSE))
```

```
actitudes_lenguaje_post$Afectivo_post = apply(actitudes_lenguaje_post[,c(7, 10, 13, 16)], 1,
                                              function(x) sum(x, na.rm= FALSE))
```

Cognitivo

```
actitudes_lenguaje_pre$Cognitivo_pre = apply(actitudes_lenguaje_pre[,c(8, 11, 14, 17)], 1,
                                              function(x) sum(x, na.rm= FALSE))
```

```
actitudes_lenguaje_post$Cognitivo_post = apply(actitudes_lenguaje_post[,c(8, 11, 14, 17)], 1,
                                                function(x) sum(x, na.rm= FALSE))
```

```

# Conativo

actitudes_lenguaje_pre$Conativo_pre = apply(actitudes_lenguaje_pre[,c(9, 12, 15, 18)], 1,
      function(x) sum(x, na.rm= FALSE))
actitudes_lenguaje_post$Conativo_post = apply(actitudes_lenguaje_post[,c(9, 12, 15, 18)], 1,
      function(x) sum(x, na.rm= FALSE))

# Matriz pre y post para comparación de muestras

pre_post = inner_join(actitudes_lenguaje_post,
      dplyr::select(actitudes_lenguaje_pre, c("Código",
      "Total_pre",
      "Afectivo_pre",
      "Cognitivo_pre",
      "Conativo_pre")),
      by = "Código")

```

Iremos guardando los resultados de cada estudiante en una base aparte

```

pivot_to_bind = dplyr::select(pre_post, c("Código", "Total_pre",
      "Afectivo_pre",
      "Cognitivo_pre",
      "Conativo_pre",
      "Total_post", "Afectivo_post",
      "Cognitivo_post",
      "Conativo_post"))

pivot_to_bind[,2:ncol(pivot_to_bind)] =
  apply(pivot_to_bind[,2:ncol(pivot_to_bind)],
    2, function(x) min_max_scale(x))

pivot_to_bind =
  pivot_to_bind %>%
  pivot_longer(cols = !`Código`,
    names_to = c("Prueba", "Tipo"),
    names_sep = "_",
    values_to = "score")

pivot_to_bind$Prueba = str_replace_all(pivot_to_bind$Prueba,
  c("Total" = "Actitudes hacia el lenguaje - Total",
    "Afectivo" = "Actitudes hacia el lenguaje - Afectivo",
    "Cognitivo" = "Actitudes hacia el lenguaje - Cognitivo",
    "Conativo" = "Actitudes hacia el lenguaje - Conativo"))

pivot_final = pivot_to_bind

```

Estadísticos psicométricos

3.1.1.1 Alpha

```
alpha(actitudes_lenguaje_pre[,7:18])
```

3.1.1.1.1 Total pre

```
## [1] 0.78413
```

```
for(i in seq(length(colnames(actitudes_lenguaje_pre[,7:18])))){  
  x = alpha(actitudes_lenguaje_pre[,7:18][,-i])  
  print(paste("El índice de confiabilidad cambia a", round(x, 2),  
             "al eliminar el ítem",  
             colnames(actitudes_lenguaje_pre[,7:18])[i]))  
}
```

```
## [1] "El índice de confiabilidad cambia a 0.76 al eliminar el ítem Grupo 1 _ 1"  
## [1] "El índice de confiabilidad cambia a 0.77 al eliminar el ítem Grupo 1 _ 2"  
## [1] "El índice de confiabilidad cambia a 0.77 al eliminar el ítem Grupo 1 _ 3"  
## [1] "El índice de confiabilidad cambia a 0.77 al eliminar el ítem Grupo 2 _ 1"  
## [1] "El índice de confiabilidad cambia a 0.76 al eliminar el ítem Grupo 2 _ 2"  
## [1] "El índice de confiabilidad cambia a 0.76 al eliminar el ítem Grupo 2 _ 3"  
## [1] "El índice de confiabilidad cambia a 0.77 al eliminar el ítem Grupo 3 _ 1"  
## [1] "El índice de confiabilidad cambia a 0.77 al eliminar el ítem Grupo 3 _ 2"  
## [1] "El índice de confiabilidad cambia a 0.82 al eliminar el ítem Grupo 3 _ 3"  
## [1] "El índice de confiabilidad cambia a 0.76 al eliminar el ítem Grupo 4 _ 1"  
## [1] "El índice de confiabilidad cambia a 0.76 al eliminar el ítem Grupo 4 _ 2"  
## [1] "El índice de confiabilidad cambia a 0.75 al eliminar el ítem Grupo 4 _ 3"
```

```
alpha(actitudes_lenguaje_post[,7:18])
```

3.1.1.1.2 Total post

```
## [1] 0.85479
```

```
for(i in seq(length(colnames(actitudes_lenguaje_post[,7:18])))){  
  x = alpha(actitudes_lenguaje_post[,7:18][,-i])  
  print(paste("El índice de confiabilidad cambia a", round(x, 2),  
             "al eliminar el ítem",  
             colnames(actitudes_lenguaje_post[,7:18])[i]))  
}
```

```
## [1] "El índice de confiabilidad cambia a 0.84 al eliminar el ítem Grupo 1 _ 1"  
## [1] "El índice de confiabilidad cambia a 0.84 al eliminar el ítem Grupo 1 _ 2"  
## [1] "El índice de confiabilidad cambia a 0.85 al eliminar el ítem Grupo 1 _ 3"  
## [1] "El índice de confiabilidad cambia a 0.85 al eliminar el ítem Grupo 2 _ 1"  
## [1] "El índice de confiabilidad cambia a 0.85 al eliminar el ítem Grupo 2 _ 2"  
## [1] "El índice de confiabilidad cambia a 0.84 al eliminar el ítem Grupo 2 _ 3"  
## [1] "El índice de confiabilidad cambia a 0.84 al eliminar el ítem Grupo 3 _ 1"  
## [1] "El índice de confiabilidad cambia a 0.84 al eliminar el ítem Grupo 3 _ 2"
```

```
## [1] "El índice de confiabilidad cambia a 0.84 al eliminar el ítem Grupo 3 _ 3"
## [1] "El índice de confiabilidad cambia a 0.84 al eliminar el ítem Grupo 4 _ 1"
## [1] "El índice de confiabilidad cambia a 0.84 al eliminar el ítem Grupo 4 _ 2"
## [1] "El índice de confiabilidad cambia a 0.84 al eliminar el ítem Grupo 4 _ 3"
```

```
# Creamos una base para guardar los valores alpha generales
```

```
alfa = data.frame(matrix(ncol = 2))
colnames(alfa) = c("Prueba", "Alfa")
```

```
# Amacemos los valores
```

```
alfa = rbind(alfa,
              c("Actitudes lenguaje pre",
                alpha(actitudes_lenguaje_pre[,7:18])))

alfa = rbind(alfa,
              c("Actitudes lenguaje post",
                alpha(actitudes_lenguaje_post[,7:18])))
```

```
alpha(actitudes_lenguaje_pre[,c(8, 11, 14, 17)])
```

3.1.1.1.3 Componente cognitivo pre

```
## [1] 0.60861
```

```
for(i in seq(length(colnames(actitudes_lenguaje_pre[,c(8, 11, 14, 17)])))){
  x = alpha(actitudes_lenguaje_pre[,c(8, 11, 14, 17)][,-i])
  print(paste("El índice de confiabilidad cambia a", x, "al eliminar el ítem",
              colnames(actitudes_lenguaje_pre[,c(8, 11, 14, 17)])[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.556694321572943 al eliminar el ítem Grupo 1 _ 2"
## [1] "El índice de confiabilidad cambia a 0.498430232915343 al eliminar el ítem Grupo 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.538203044884436 al eliminar el ítem Grupo 3 _ 2"
## [1] "El índice de confiabilidad cambia a 0.55922017601043 al eliminar el ítem Grupo 4 _ 2"
```

```
alpha(actitudes_lenguaje_post[,c(8, 11, 14, 17)])
```

3.1.1.1.4 Componente cognitivo post

```
## [1] 0.62034
```

```
for(i in seq(length(colnames(actitudes_lenguaje_post[,c(8, 11, 14, 17)])))){
  x = alpha(actitudes_lenguaje_post[,c(8, 11, 14, 17)][,-i])
  print(paste("El índice de confiabilidad cambia a", x, "al eliminar el ítem",
              colnames(actitudes_lenguaje_post[,c(8, 11, 14, 17)])[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.543958370762005 al eliminar el ítem Grupo 1 _ 2"
## [1] "El índice de confiabilidad cambia a 0.540325580798686 al eliminar el ítem Grupo 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.579548216687462 al eliminar el ítem Grupo 3 _ 2"
## [1] "El índice de confiabilidad cambia a 0.544770523594053 al eliminar el ítem Grupo 4 _ 2"
```

```
alpha(actitudes_lenguaje_pre[,c(7, 10, 13, 16)])
```

3.1.1.1.5 Componente afectivo pre

```
## [1] 0.60421
```

```
for(i in seq(length(colnames(actitudes_lenguaje_pre[,c(7, 10, 13, 16)])))){
  x = alpha(actitudes_lenguaje_pre[,c(7, 10, 13, 16)][,-i])
  print(paste("El índice de confiabilidad cambia a", x, "al eliminar el ítem",
             colnames(actitudes_lenguaje_pre[,c(7, 10, 13, 16)])[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.494813278008299 al eliminar el ítem Grupo 1 _ 1"
## [1] "El índice de confiabilidad cambia a 0.52592850700992 al eliminar el ítem Grupo 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.576818485771603 al eliminar el ítem Grupo 3 _ 1"
## [1] "El índice de confiabilidad cambia a 0.524590501699804 al eliminar el ítem Grupo 4 _ 1"
```

```
alpha(actitudes_lenguaje_post[,c(7, 10, 13, 16)])
```

3.1.1.1.6 Componente afectivo post

```
## [1] 0.62652
```

```
for(i in seq(length(colnames(actitudes_lenguaje_post[,c(7, 10, 13, 16)])))){
  x = alpha(actitudes_lenguaje_post[,c(7, 10, 13, 16)][,-i])
  print(paste("El índice de confiabilidad cambia a", x, "al eliminar el ítem",
             colnames(actitudes_lenguaje_post[,c(7, 10, 13, 16)])[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.543114470939231 al eliminar el ítem Grupo 1 _ 1"
## [1] "El índice de confiabilidad cambia a 0.560477562783038 al eliminar el ítem Grupo 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.583479020979021 al eliminar el ítem Grupo 3 _ 1"
## [1] "El índice de confiabilidad cambia a 0.545571909691905 al eliminar el ítem Grupo 4 _ 1"
```

```
alpha(actitudes_lenguaje_pre[,c(9, 12, 15, 18)])
```

3.1.1.1.7 Componente conativo pre

```
## [1] 0.38942
```

```
for(i in seq(length(colnames(actitudes_lenguaje_pre[,c(9, 12, 15, 18)])))){
  x = alpha(actitudes_lenguaje_pre[,c(9, 12, 15, 18)][,-i])
  print(paste("El índice de confiabilidad cambia a", x, "al eliminar el ítem",
```

```
colnames(actitudes_lenguaje_pre[,c(9, 12, 15, 18))][i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.180439508185585 al eliminar el ítem Grupo 1 _ 3"
## [1] "El índice de confiabilidad cambia a 0.188469241925972 al eliminar el ítem Grupo 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.557564695009242 al eliminar el ítem Grupo 3 _ 3"
## [1] "El índice de confiabilidad cambia a 0.323961671512553 al eliminar el ítem Grupo 4 _ 3"
```

```
alpha(actitudes_lenguaje_post[,c(9, 12, 15, 18)])
```

3.1.1.1.8 Componente conativo post

```
## [1] 0.62241
```

```
for(i in seq(length(colnames(actitudes_lenguaje_post[,c(9, 12, 15, 18)))])){
  x = alpha(actitudes_lenguaje_post[,c(9, 12, 15, 18)][,-i])
  print(paste("El índice de confiabilidad cambia a", x, "al eliminar el ítem",
             colnames(actitudes_lenguaje_post[,c(9, 12, 15, 18))][i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.534619310565346 al eliminar el ítem Grupo 1 _ 3"
## [1] "El índice de confiabilidad cambia a 0.534452276536634 al eliminar el ítem Grupo 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.607236057262153 al eliminar el ítem Grupo 3 _ 3"
## [1] "El índice de confiabilidad cambia a 0.538874858954382 al eliminar el ítem Grupo 4 _ 3"
```

3.1.1.2 Indicadores psicométricos

```
analitem = item.exam(actitudes_lenguaje_pre[,7:18], discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    )
  )
```

```
),
locations = cells_body())
```

3.1.1.2.1 Total pre

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(cols_num)` instead of `cols_num` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.387	0.604	0.492	0.818	0.412	0.233
0.308	0.545	0.450	0.894	0.291	0.168
0.391	0.530	0.405	0.812	0.412	0.207
0.448	0.589	0.454	0.723	0.577	0.264
0.359	0.618	0.518	0.849	0.401	0.222
0.350	0.635	0.541	0.858	0.379	0.222
0.266	0.538	0.457	0.923	0.214	0.143
0.291	0.543	0.454	0.907	0.253	0.158
0.467	0.132	-0.053	0.319	0.170	0.062
0.450	0.619	0.489	0.719	0.637	0.278
0.428	0.639	0.521	0.759	0.593	0.273
0.422	0.684	0.578	0.768	0.577	0.289

```
# Creamos una base para guardar indicadores
```

```
indicadores_psicometricos = data.frame()
```

```
# Guardamos los indicadores importantes
```

```
psicometricos_pre = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Actitudes hacia el lenguaje - Pre") %>%
  mutate(Item = row.names(.))
```

```
indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_pre)
```

```
analitem = item.exam(actitudes_lenguaje_post[,7:18], discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
```

```

    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body()
  )

```

3.1.1.2.2 Total post

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.293	0.598	0.519	0.906	0.266	0.175
0.306	0.613	0.533	0.896	0.295	0.187
0.382	0.603	0.498	0.823	0.422	0.230
0.336	0.596	0.504	0.871	0.335	0.200
0.303	0.560	0.473	0.898	0.283	0.169
0.344	0.632	0.544	0.863	0.370	0.217
0.426	0.643	0.532	0.763	0.601	0.273
0.426	0.677	0.574	0.763	0.642	0.288
0.423	0.656	0.549	0.767	0.613	0.277
0.360	0.661	0.573	0.848	0.422	0.237
0.311	0.609	0.526	0.892	0.301	0.189
0.303	0.635	0.558	0.898	0.277	0.192

```

# Guardamos los indicadores importantes

psicometricos_post = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Actitudes hacia el lenguaje - Post") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_post)

```

3.1.1.3 Comparación pre-post Pese a que la mayoría de las pruebas conservan el mismo número de preguntas finales, no es el caso para todas. Es por eso que, con el fin de permitir la comparación entre las pruebas pre y post, haremos un escalamiento min-max tal que todas las puntuaciones queden entre 0 y 1.


```
pre_post$Total_pre = min_max_scale(pre_post$Total_pre)
pre_post$Total_post = min_max_scale(pre_post$Total_post)
```

3.1.1.3.1 Prueba total Estadísticos de normalidad

```
print("Estadístico de normalidad pre")
```

```
## [1] "Estadístico de normalidad pre"
```

```
lillie.test(pre_post$Total_pre)
```

```
##
```

```
## Lilliefors (Kolmogorov-Smirnov) normality test
```

```
##
```

```
## data: pre_post$Total_pre
```

```
## D = 0.225, p-value <0.0000000000000002
```

```
print("Estadístico de normalidad post")
```

```
## [1] "Estadístico de normalidad post"
```

```
lillie.test(pre_post$Total_post)
```

```
##
```

```
## Lilliefors (Kolmogorov-Smirnov) normality test
```

```
##
```

```
## data: pre_post$Total_post
```

```
## D = 0.268, p-value <0.0000000000000002
```

Debido a que no existe normalidad en las puntuaciones, se usará una prueba no paramétrica

Descriptivos

```
summ = data.frame(unclass(psych::describe(pre_post[,c("Total_pre", "Total_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)
```

```
summ$vars = c("Total_pre", "Total_post")
```

```
summ[,1:10] %>%
```

```
gt()
```

vars	n	mean	sd	median	trimmed	mad	min	max	range
Total_pre	433	0.77868	0.21199	0.83333	0.81124	0.12355	0	1	1
Total_post	433	0.85758	0.21145	1.00000	0.90322	0.00000	0	1	1

Comparación de medias

```
comparacion =
  wilcox.test(
    x = pre_post$Total_pre,
```

```

y          = pre_post$Total_post,
alternative = "two.sided",
mu         = 0,
var.equal  = TRUE,
paired     = TRUE,
conf.level = 0.95
)

```

comparacion

```

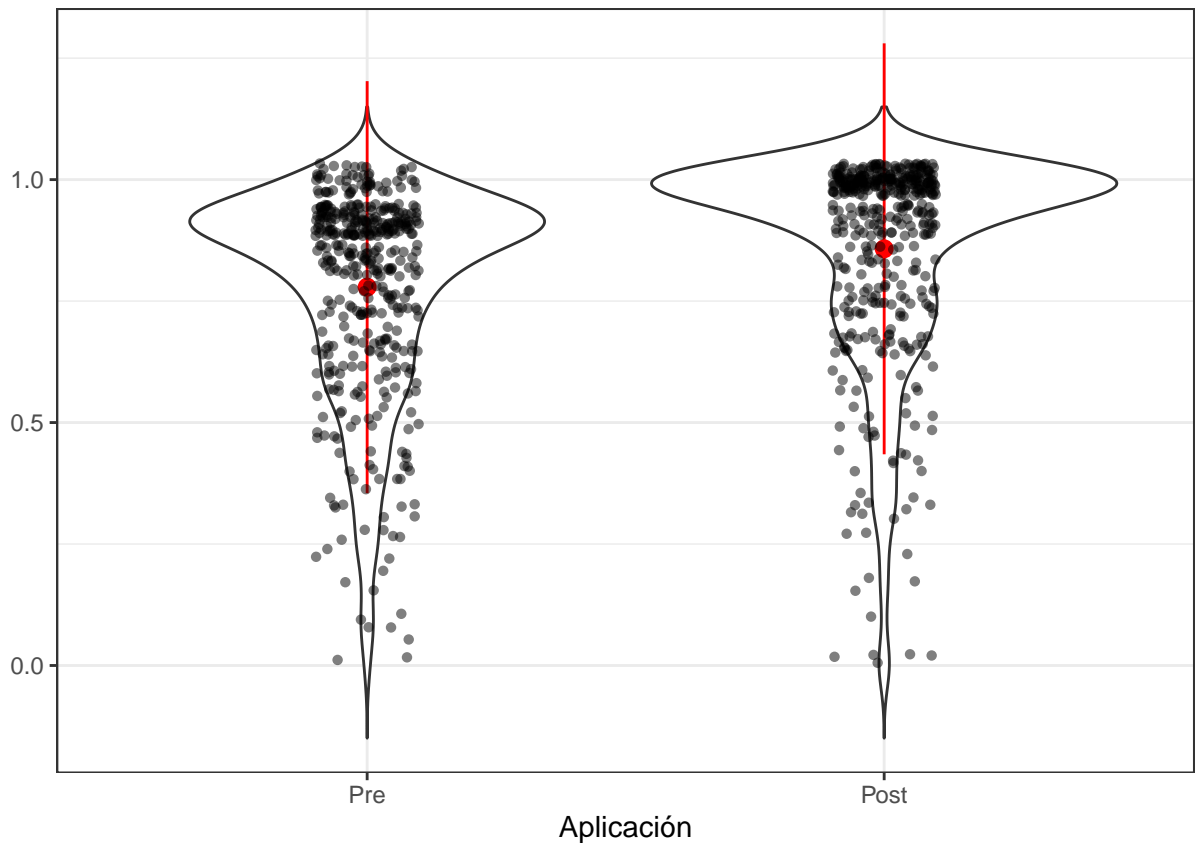
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Total_pre and pre_post$Total_post
## V = 14610, p-value <0.0000000000000002
## alternative hypothesis: true location shift is not equal to 0

```

```

pre_post %>%
  dplyr::select(Total_pre, Total_post) %>%
  pivot_longer(cols = c(Total_pre, Total_post),
               values_to = "value",
               names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
  scale_x_discrete(labels = c("Pre", "Post"))

```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/actitudes_lenguaje.png")
```

Tamaño del efecto

```
size_effect =  
  cohen.d(pre_post$Total_post, pre_post$Total_pre, paired = TRUE)  
  
size_effect
```

```
##  
## Cohen's d  
##  
## d estimate: 0.37269 (small)  
## 95 percent confidence interval:  
##   lower   upper  
## 0.27135 0.47403
```

```
# Creamos una base para ir guardando los descriptivo
```

```
descriptivos = data.frame()
```

```
# Separamos los descriptivos importantes
```

```
desc = summ[,1:10] %>%
```

```

dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
  mutate(Prueba = "Actitudes hacia el lenguaje - Total") %>%
  mutate(Area = "Actitudes hacia el lenguaje") %>%
  mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Adicioalmente, guardaremos en otra tabla las comparaciones

comparaciones = data.frame()

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Actitudes hacia el lenguaje - Total",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)

```

```

pre_post$Afectivo_pre = min_max_scale(pre_post$Afectivo_pre)
pre_post$Afectivo_post = min_max_scale(pre_post$Afectivo_post)

```

3.1.1.3.2 Componente afectivo Estadísticos de normalidad

```

print("Estadístico de normalidad pre")

## [1] "Estadístico de normalidad pre"
lillie.test(pre_post$Afectivo_pre)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Afectivo_pre
## D = 0.305, p-value <0.0000000000000002
print("Estadístico de normalidad post")

## [1] "Estadístico de normalidad post"
lillie.test(pre_post$Afectivo_post)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Afectivo_post

```

```
## D = 0.377, p-value <0.0000000000000002
```

Debido a que no existe normalidad en las puntuaciones, se usará una prueba no paramétrica

Descriptivos

```
summ = data.frame(unclass(psych::describe(pre_post[,c("Afectivo_pre",
                                                    "Afectivo_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Afectivo_pre", "Afectivo_post")

summ[,1:10] %>%
  gt()
```

vars	n	mean	sd	median	trimmed	mad	min	max	range
Afectivo_pre	433	0.79561	0.26423	1	0.84294	0	0	1	1
Afectivo_post	433	0.85566	0.23498	1	0.90490	0	0	1	1

Comparación de medias

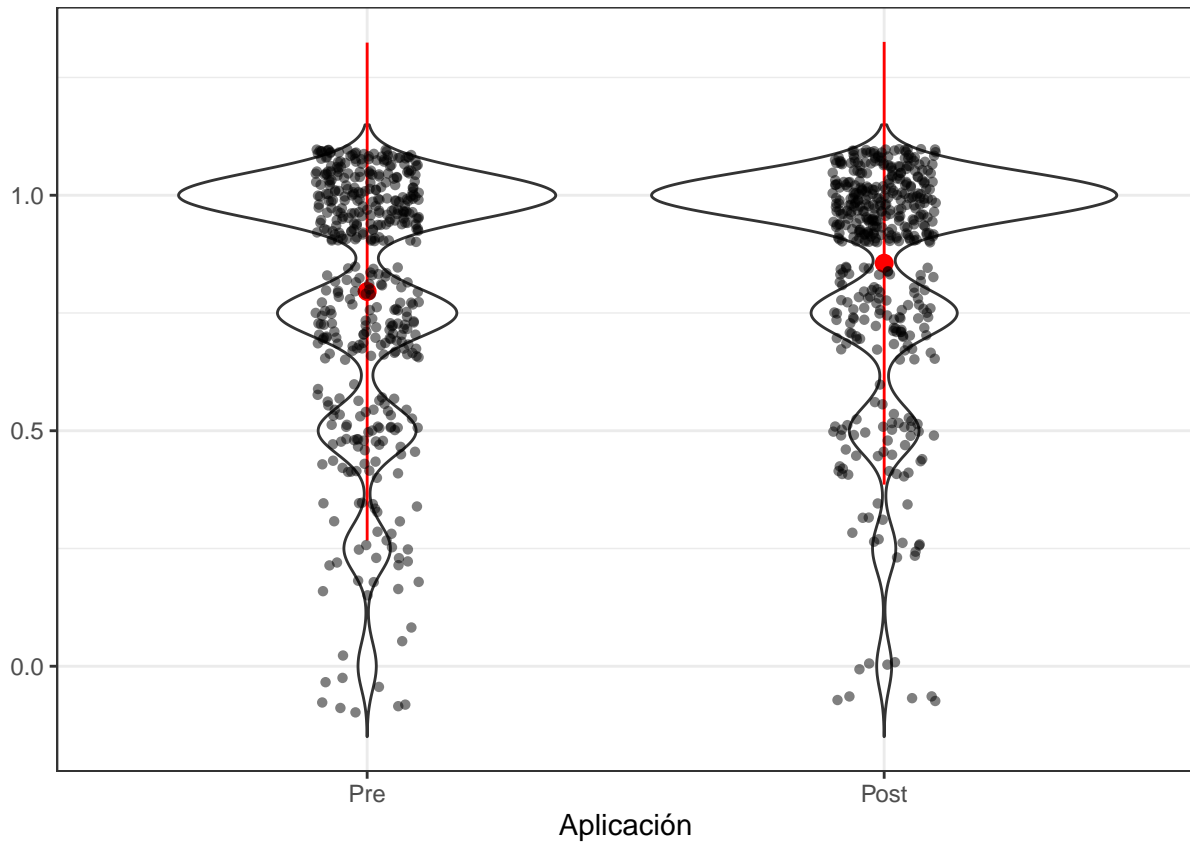
```
comparacion =
  wilcox.test(
    x      = pre_post$Afectivo_pre,
    y      = pre_post$Afectivo_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )
```

```
comparacion
```

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Afectivo_pre and pre_post$Afectivo_post
## V = 6312, p-value = 0.0000033
## alternative hypothesis: true location shift is not equal to 0
```

```
pre_post %>%
  dplyr::select(Afectivo_pre, Afectivo_post) %>%
  pivot_longer(cols = c(Afectivo_pre, Afectivo_post),
               values_to = "value",
               names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
```

```
theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
  scale_x_discrete(labels = c("Pre", "Post"))
```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/actitudes_lenguaje_afectivo.png")
```

Tamaño del efecto

```
size_effect =
  cohen.d(pre_post$Afectivo_post, pre_post$Afectivo_pre, paired = TRUE)

size_effect
```

```
##
## Cohen's d
##
## d estimate: 0.23958 (small)
## 95 percent confidence interval:
##   lower   upper
## 0.13559 0.34357
```

```

# Separamos los descriptivos importantes
desc = summ[,1:10] %>%
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
  mutate(Prueba = "Actitudes hacia el lenguaje - Afectivo") %>%
  mutate(Area = "Actitudes hacia el lenguaje") %>%
  mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Actitudes hacia el lenguaje - Afectivo",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)

```

```

pre_post$Cognitivo_pre = min_max_scale(pre_post$Cognitivo_pre)
pre_post$Cognitivo_post = min_max_scale(pre_post$Cognitivo_post)

```

3.1.1.3.3 Componente Cognitivo Estadísticos de normalidad

```

print("Estadístico de normalidad pre")

## [1] "Estadístico de normalidad pre"
lillie.test(pre_post$Cognitivo_pre)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Cognitivo_pre
## D = 0.375, p-value <0.0000000000000002

print("Estadístico de normalidad post")

## [1] "Estadístico de normalidad post"
lillie.test(pre_post$Cognitivo_post)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Cognitivo_post
## D = 0.399, p-value <0.0000000000000002

```

Debido a que no existe normalidad en las puntuaciones, se usará una prueba no paramétrica

Descriptivos

```
summ = data.frame(unclass(psych::describe(pre_post[,c("Cognitivo_pre",
                                                    "Cognitivo_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Cognitivo_pre", "Cognitivo_post")

summ[,1:10] %>%
  gt()
```

vars	n	mean	sd	median	trimmed	mad	min	max	range
Cognitivo_pre	433	0.85277	0.23411	1	0.90130	0	0	1	1
Cognitivo_post	433	0.87298	0.21791	1	0.92075	0	0	1	1

Comparación de medias

```
comparacion =
  wilcox.test(
    x      = pre_post$Cognitivo_pre,
    y      = pre_post$Cognitivo_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )
```

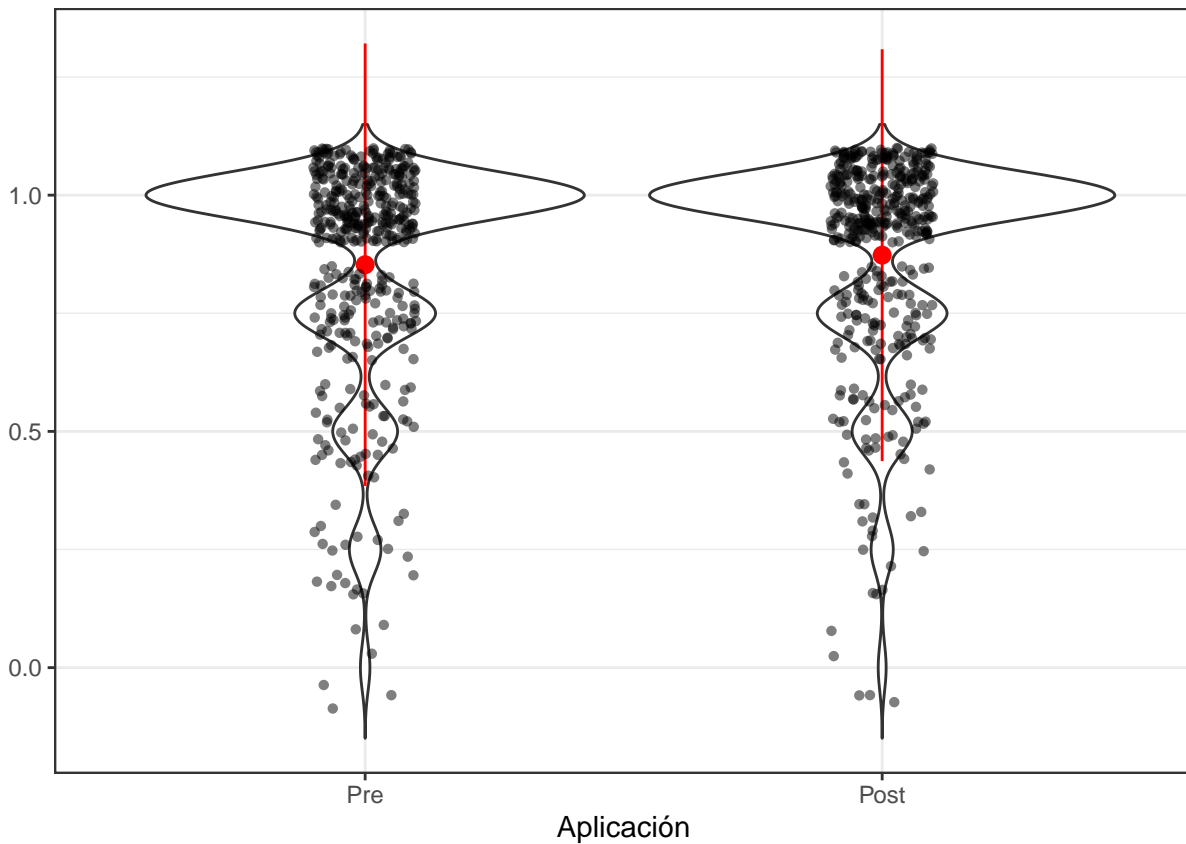
```
comparacion
```

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Cognitivo_pre and pre_post$Cognitivo_post
## V = 6728, p-value = 0.079
## alternative hypothesis: true location shift is not equal to 0
```

```
pre_post %>%
  dplyr::select(Cognitivo_pre, Cognitivo_post) %>%
  pivot_longer(cols = c(Cognitivo_pre, Cognitivo_post),
               values_to = "value",
               names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +
```



```
geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
scale_x_discrete(labels = c("Pre", "Post"))
```



Guardamos la imagen

```
ggsave("../Plots/segunda_aplicacion/actitudes_lenguaje_cognitivo.png")
```

Tamaño del efecto

```
size_effect =
  cohen.d(pre_post$Cognitivo_post, pre_post$Cognitivo_pre, paired = TRUE)
```

```
size_effect
```

```
##
## Cohen's d
##
## d estimate: 0.089285 (negligible)
## 95 percent confidence interval:
##    lower    upper
## -0.016331 0.194900
```

Separamos los descriptivos importantes

```
desc = summ[,1:10] %>%
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
```

```

mutate(Prueba = "Actitudes hacia el lenguaje - Cognitivo") %>%
mutate(Area = "Actitudes hacia el lenguaje") %>%
mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Actitudes hacia el lenguaje - Cognitivo",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)

```

```

pre_post$Conativo_pre = min_max_scale(pre_post$Conativo_pre)
pre_post$Conativo_post = min_max_scale(pre_post$Conativo_post)

```

3.1.1.3.4 Componente Conativo Estadísticos de normalidad

```

print("Estadístico de normalidad pre")

## [1] "Estadístico de normalidad pre"
lillie.test(pre_post$Conativo_pre)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Conativo_pre
## D = 0.312, p-value <0.0000000000000002
print("Estadístico de normalidad post")

## [1] "Estadístico de normalidad post"
lillie.test(pre_post$Conativo_post)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Conativo_post
## D = 0.361, p-value <0.0000000000000002

```

Debido a que no existe normalidad en las puntuaciones, se usará una prueba no paramétrica

Descriptivos

```
summ = data.frame(unclass(psych::describe(pre_post[,c("Conativo_pre",
                                                    "Conativo_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Conativo_pre", "Conativo_post")

summ[,1:10] %>%
  gt()
```

vars	n	mean	sd	median	trimmed	mad	min	max	range
Conativo_pre	433	0.68764	0.24564	0.75	0.71542	0	0	1	1
Conativo_post	433	0.84411	0.24522	1.00	0.89481	0	0	1	1

Comparación de medias

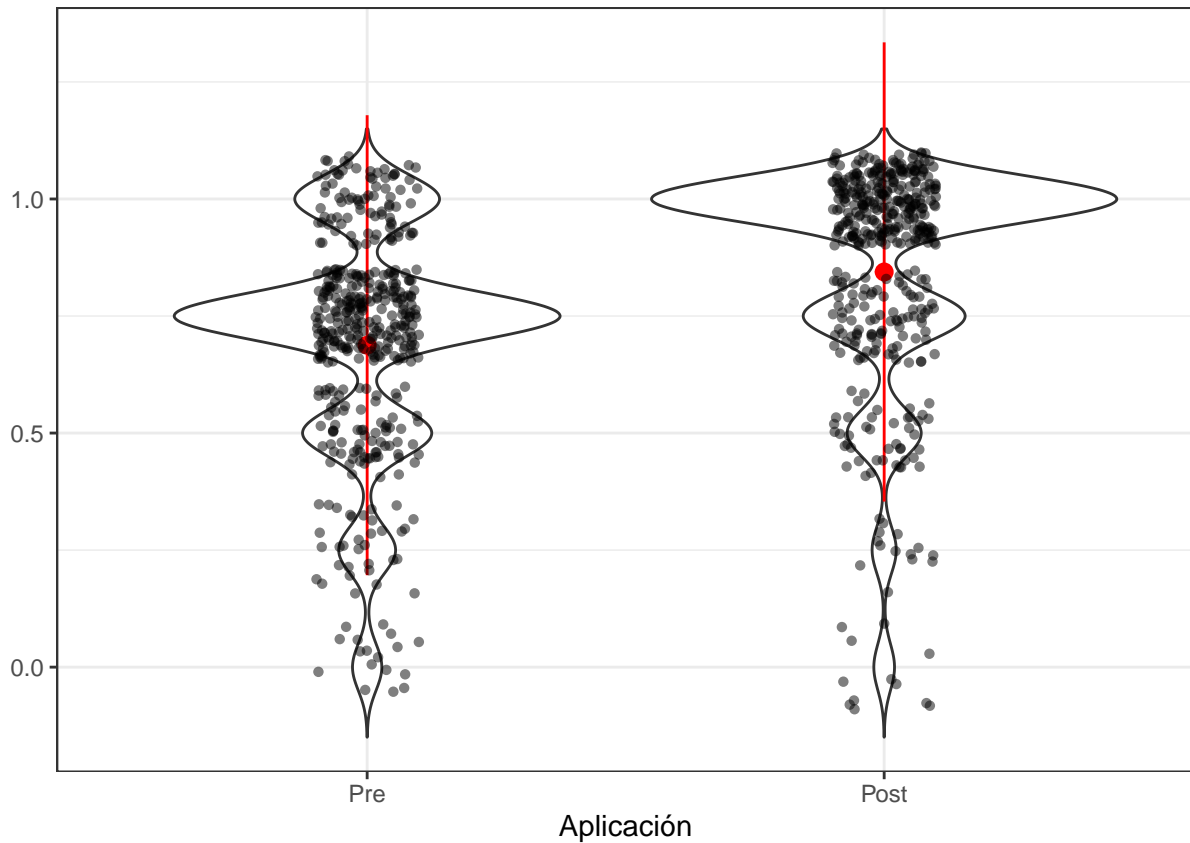
```
comparacion =
  wilcox.test(
    x      = pre_post$Conativo_pre,
    y      = pre_post$Conativo_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )
```

```
comparacion
```

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Conativo_pre and pre_post$Conativo_post
## V = 9258, p-value <0.0000000000000002
## alternative hypothesis: true location shift is not equal to 0
```

```
pre_post %>%
  dplyr::select(Conativo_pre, Conativo_post) %>%
  pivot_longer(cols = c(Conativo_pre, Conativo_post),
               values_to = "value",
               names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
```

```
scale_x_discrete(labels = c("Pre", "Post"))
```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/actitudes_lenguaje_conativo.png")
```

Tamaño del efecto

```
size_effect =  
  cohen.d(pre_post$Conativo_post, pre_post$Conativo_pre, paired = TRUE)
```

```
size_effect
```

```
##  
## Cohen's d  
##  
## d estimate: 0.63752 (medium)  
## 95 percent confidence interval:  
##   lower   upper  
## 0.51824 0.75680
```

```
# Separamos los descriptivos importantes
```

```
desc = summ[,1:10] %>%  
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%  
  mutate(Prueba = "Actitudes hacia el lenguaje - Conativo") %>%
```

```

mutate(Area = "Actitudes hacia el lenguaje") %>%
mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Actitudes hacia el lenguaje - Conativo",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)

```

3.1.2 Actitudes hacia las matemáticas

```

actitudes_matematicas_pre = filter(actitudes_pre,
  substr(`Código`, 1, 1) == "2")
actitudes_matematicas_post = filter(actitudes_post,
  substr(`Código`, 1, 1) == "2")

```

Calificación pre

```

actitudes_matematicas_pre[,7:18] =
  calificacion(actitudes_matematicas_pre[,7:18],
    claves_matematicas_pre)

```

Calificación post

```

actitudes_matematicas_post[,7:18] =
  calificacion(actitudes_matematicas_post[,7:18],
    claves_matematicas_post)

```

Calificación

Total

```

actitudes_matematicas_pre$Total_pre =
  apply(actitudes_matematicas_pre[,7:18], 1,
    function(x) sum(x, na.rm= FALSE))

```

```

actitudes_matematicas_post$Total_post =
  apply(actitudes_matematicas_post[,7:18], 1,
    function(x) sum(x, na.rm= FALSE))

```

Afectivo

```

actitudes_matematicas_pre$Afectivo_pre =
  apply(actitudes_matematicas_pre[,c(7, 10, 13, 16)], 1,
        function(x) sum(x, na.rm= FALSE))

actitudes_matematicas_post$Afectivo_post =
  apply(actitudes_matematicas_post[,c(7, 10, 13, 16)], 1,
        function(x) sum(x, na.rm= FALSE))

# Cognitivo

actitudes_matematicas_pre$Cognitivo_pre =
  apply(actitudes_matematicas_pre[,c(8, 11, 14, 17)], 1,
        function(x) sum(x, na.rm= FALSE))

actitudes_matematicas_post$Cognitivo_post =
  apply(actitudes_matematicas_post[,c(8, 11, 14, 17)], 1,
        function(x) sum(x, na.rm= FALSE))

# Conativo

actitudes_matematicas_pre$Conativo_pre =
  apply(actitudes_matematicas_pre[,c(9, 12, 15, 18)], 1,
        function(x) sum(x, na.rm= FALSE))

actitudes_matematicas_post$Conativo_post =
  apply(actitudes_matematicas_post[,c(9, 12, 15, 18)], 1,
        function(x) sum(x, na.rm= FALSE))

pre_post = inner_join(actitudes_matematicas_post,
                      dplyr::select(actitudes_matematicas_pre,
                                    c("Código",
                                      "Total_pre",
                                      "Afectivo_pre",
                                      "Cognitivo_pre",
                                      "Conativo_pre")),
                      by = "Código")

```

3.1.2.1 Alpha

```
alpha(actitudes_matematicas_pre[,7:18])
```

3.1.2.1.1 Total pre

```
## [1] 0.79767
```

```

for(i in seq(length(colnames(actitudes_matematicas_pre[,7:18])))){
  x = alpha(actitudes_matematicas_pre[,7:18][,-i])
}

```

```
print(paste("El índice de confiabilidad cambia a", x,
            "al eliminar el ítem",
            colnames(actitudes_matematicas_pre[,7:18])[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.781074547641002 al eliminar el ítem Grupo 1 _ 1"
## [1] "El índice de confiabilidad cambia a 0.788742691990091 al eliminar el ítem Grupo 1 _ 2"
## [1] "El índice de confiabilidad cambia a 0.775035987440339 al eliminar el ítem Grupo 1 _ 3"
## [1] "El índice de confiabilidad cambia a 0.789840510132492 al eliminar el ítem Grupo 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.78094127257564 al eliminar el ítem Grupo 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.773399161863011 al eliminar el ítem Grupo 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.784933825666217 al eliminar el ítem Grupo 3 _ 1"
## [1] "El índice de confiabilidad cambia a 0.780922891196617 al eliminar el ítem Grupo 3 _ 2"
## [1] "El índice de confiabilidad cambia a 0.787956064126658 al eliminar el ítem Grupo 3 _ 3"
## [1] "El índice de confiabilidad cambia a 0.790010785513887 al eliminar el ítem Grupo 4 _ 1"
## [1] "El índice de confiabilidad cambia a 0.776636775204597 al eliminar el ítem Grupo 4 _ 2"
## [1] "El índice de confiabilidad cambia a 0.788660678884713 al eliminar el ítem Grupo 4 _ 3"
```

```
alpha(actitudes_matematicas_post[,7:18])
```

3.1.2.1.2 Total post

```
## [1] 0.75239
```

```
for(i in seq(length(colnames(actitudes_matematicas_post[,7:18])))){
  x = alpha(actitudes_matematicas_post[,7:18][,-i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(actitudes_matematicas_post[,7:18])[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.722737045564512 al eliminar el ítem Grupo 1 _ 1"
## [1] "El índice de confiabilidad cambia a 0.717871537467114 al eliminar el ítem Grupo 1 _ 2"
## [1] "El índice de confiabilidad cambia a 0.720937866651446 al eliminar el ítem Grupo 1 _ 3"
## [1] "El índice de confiabilidad cambia a 0.729298279601393 al eliminar el ítem Grupo 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.727917262957453 al eliminar el ítem Grupo 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.730971313006124 al eliminar el ítem Grupo 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.824724982232701 al eliminar el ítem Grupo 3 _ 1"
## [1] "El índice de confiabilidad cambia a 0.728767314957325 al eliminar el ítem Grupo 3 _ 2"
## [1] "El índice de confiabilidad cambia a 0.719173058185725 al eliminar el ítem Grupo 3 _ 3"
## [1] "El índice de confiabilidad cambia a 0.723855743266418 al eliminar el ítem Grupo 4 _ 1"
## [1] "El índice de confiabilidad cambia a 0.72730097370972 al eliminar el ítem Grupo 4 _ 2"
## [1] "El índice de confiabilidad cambia a 0.725377182573318 al eliminar el ítem Grupo 4 _ 3"
```

```
# Amacemos los valores
```

```
alfa = rbind(alfa,
             c("Actitudes matemáticas pre",
               alpha(actitudes_matematicas_pre[,7:18])))
```

```

alfa = rbind(alfa,
             c("Actitudes matemáticas post",
               alpha(actitudes_matematicas_post[,7:18])))

```

```

alpha(actitudes_matematicas_pre[,c(8, 11, 14, 17)])

```

3.1.2.1.3 Componente cognitivo pre

```
## [1] 0.58756
```

```

for(i in seq(length(colnames(actitudes_matematicas_pre[,c(8, 11, 14, 17)]))){
  x = alpha(actitudes_matematicas_pre[,c(8, 11, 14, 17)][,-i])
  print(paste("El índice de confiabilidad cambia a", x, "al eliminar el ítem",
              colnames(actitudes_matematicas_pre[,c(8, 11, 14, 17)])[i]))
}

```

```

## [1] "El índice de confiabilidad cambia a 0.546930724317501 al eliminar el ítem Grupo 1 _ 2"
## [1] "El índice de confiabilidad cambia a 0.468731657862441 al eliminar el ítem Grupo 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.531060811421343 al eliminar el ítem Grupo 3 _ 2"
## [1] "El índice de confiabilidad cambia a 0.513035446655746 al eliminar el ítem Grupo 4 _ 2"

```

```

alpha(actitudes_matematicas_post[,c(8, 11, 14, 17)])

```

3.1.2.1.4 Componente cognitivo post

```
## [1] 0.6337
```

```

for(i in seq(length(colnames(actitudes_matematicas_post[,c(8, 11, 14, 17)]))){
  x = alpha(actitudes_matematicas_post[,c(8, 11, 14, 17)][,-i])
  print(paste("El índice de confiabilidad cambia a", x, "al eliminar el ítem",
              colnames(actitudes_matematicas_post[,c(8, 11, 14, 17)])[i]))
}

```

```

## [1] "El índice de confiabilidad cambia a 0.528158014621342 al eliminar el ítem Grupo 1 _ 2"
## [1] "El índice de confiabilidad cambia a 0.625082690187431 al eliminar el ítem Grupo 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.545491369804442 al eliminar el ítem Grupo 3 _ 2"
## [1] "El índice de confiabilidad cambia a 0.550163116107224 al eliminar el ítem Grupo 4 _ 2"

```

```

alpha(actitudes_matematicas_pre[,c(7, 10, 13, 16)])

```

3.1.2.1.5 Componente afectivo pre

```
## [1] 0.54653
```

```

for(i in seq(length(colnames(actitudes_matematicas_pre[,c(7, 10, 13, 16)]))){
  x = alpha(actitudes_matematicas_pre[,c(7, 10, 13, 16)][,-i])
  print(paste("El índice de confiabilidad cambia a", x, "al eliminar el ítem",

```



```
colnames(actitudes_matematicas_pre[,c(7, 10, 13, 16))][i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.40629333933056 al eliminar el ítem Grupo 1 _ 1"
## [1] "El índice de confiabilidad cambia a 0.470731491321289 al eliminar el ítem Grupo 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.507809817214956 al eliminar el ítem Grupo 3 _ 1"
## [1] "El índice de confiabilidad cambia a 0.501507188561055 al eliminar el ítem Grupo 4 _ 1"
```

```
alpha(actitudes_matematicas_post[,c(7, 10, 13, 16)])
```

3.1.2.1.6 Componente afectivo post

```
## [1] 0.019509
```

```
for(i in seq(length(colnames(actitudes_matematicas_post[,c(7, 10, 13, 16)])))){
  x = alpha(actitudes_matematicas_post[,c(7, 10, 13, 16)][,-i])
  print(paste("El índice de confiabilidad cambia a", x, "al eliminar el ítem",
    colnames(actitudes_matematicas_post[,c(7, 10, 13, 16))][i]))
}
```

```
## [1] "El índice de confiabilidad cambia a -0.462210737680634 al eliminar el ítem Grupo 1 _ 1"
## [1] "El índice de confiabilidad cambia a -0.320382753055107 al eliminar el ítem Grupo 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.529783517948247 al eliminar el ítem Grupo 3 _ 1"
## [1] "El índice de confiabilidad cambia a -0.144005349556826 al eliminar el ítem Grupo 4 _ 1"
```

```
alpha(actitudes_matematicas_pre[,c(9, 12, 15, 18)])
```

3.1.2.1.7 Componente conativo pre

```
## [1] 0.56704
```

```
for(i in seq(length(colnames(actitudes_matematicas_pre[,c(9, 12, 15, 18)])))){
  x = alpha(actitudes_matematicas_pre[,c(9, 12, 15, 18)][,-i])
  print(paste("El índice de confiabilidad cambia a", x, "al eliminar el ítem",
    colnames(actitudes_matematicas_pre[,c(9, 12, 15, 18))][i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.511975719457344 al eliminar el ítem Grupo 1 _ 3"
## [1] "El índice de confiabilidad cambia a 0.417172698566918 al eliminar el ítem Grupo 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.47265709100975 al eliminar el ítem Grupo 3 _ 3"
## [1] "El índice de confiabilidad cambia a 0.565850824422316 al eliminar el ítem Grupo 4 _ 3"
```

```
alpha(actitudes_matematicas_post[,c(9, 12, 15, 18)])
```

3.1.2.1.8 Componente conativo post

```
## [1] 0.64433
```

```
for(i in seq(length(colnames(actitudes_matematicas_post[,c(9, 12, 15, 18)])))){
  x = alpha(actitudes_matematicas_post[,c(9, 12, 15, 18)][,-i])
  print(paste("El índice de confiabilidad cambia a", x, "al eliminar el ítem",
             colnames(actitudes_matematicas_post[,c(9, 12, 15, 18)])[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.619619443188524 al eliminar el ítem Grupo 1 _ 3"
## [1] "El índice de confiabilidad cambia a 0.60345182274117 al eliminar el ítem Grupo 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.476680965682362 al eliminar el ítem Grupo 3 _ 3"
## [1] "El índice de confiabilidad cambia a 0.593144717922693 al eliminar el ítem Grupo 4 _ 3"
```

3.1.2.2 Indicadores psicométricos

```
analitem = item.exam(actitudes_matematicas_pre[,7:18], discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body()
```

3.1.2.2.1 Total pre

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.343	0.589	0.464	0.864	0.381	0.202
0.284	0.494	0.379	0.912	0.233	0.140
0.333	0.633	0.520	0.873	0.365	0.211
0.438	0.592	0.426	0.743	0.608	0.259
0.301	0.575	0.465	0.899	0.291	0.173
0.308	0.643	0.542	0.894	0.317	0.198
0.247	0.526	0.431	0.935	0.190	0.130
0.253	0.572	0.480	0.931	0.201	0.145

0.241	0.489	0.392	0.938	0.159	0.118
0.333	0.510	0.376	0.873	0.339	0.170
0.310	0.616	0.509	0.892	0.317	0.191
0.268	0.488	0.380	0.922	0.212	0.130

```
# Guardamos los indicadores importantes
```

```
psicometricos_pre = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Actitudes hacia las matematicas - Pre") %>%
  mutate(Item = row.names(.))
```

```
indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_pre)
```

```
analitem = item.exam(actitudes_matematicas_post[,7:18], discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body()
```

3.1.2.2.2 Total post

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.341	0.621	0.484	0.866	0.401	0.211
0.270	0.647	0.548	0.921	0.236	0.174
0.333	0.629	0.498	0.874	0.379	0.209
0.255	0.561	0.453	0.930	0.198	0.143
0.245	0.573	0.471	0.936	0.192	0.140

0.291	0.553	0.426	0.907	0.280	0.161
0.315	-0.257	-0.407	0.112	-0.132	-0.081
0.289	0.568	0.445	0.908	0.247	0.164
0.304	0.635	0.518	0.897	0.297	0.193
0.320	0.608	0.478	0.885	0.313	0.194
0.239	0.580	0.482	0.940	0.181	0.138
0.283	0.592	0.476	0.912	0.258	0.168

```
# Guardamos los indicadores importantes
```

```
psicometricos_post = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Actitudes hacia las matematicas - Post") %>%
  mutate(Item = row.names(.))
```

```
indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_post)
```

Debido a que el ítem *Grupo 3 _ 1* tiene un comportamiento psicométrico indeseado, será eliminado de la calificación. Dicho ítem pertenece al componente afectivo.

```
# Calificación
```

```
# Total
```

```
actitudes_matematicas_pre$Total_pre =
  apply(actitudes_matematicas_pre[,c(7:18)[-7]], 1,
        function(x) sum(x, na.rm= FALSE))

actitudes_matematicas_post$Total_post =
  apply(actitudes_matematicas_post[,c(7:18)[-7]], 1,
        function(x) sum(x, na.rm= FALSE))
```

```
# Afectivo
```

```
actitudes_matematicas_pre$Afectivo_pre =
  apply(actitudes_matematicas_pre[,c(7, 10, 16)], 1,
        function(x) sum(x, na.rm= FALSE))

actitudes_matematicas_post$Afectivo_post =
  apply(actitudes_matematicas_post[,c(7, 10, 16)], 1,
        function(x) sum(x, na.rm= FALSE))
```

```
# Volvemos a obtener la matriz pre post
```

```
pre_post = inner_join(actitudes_matematicas_post,
                      dplyr::select(actitudes_matematicas_pre,
                                    c("Código",
```

```

                                "Total_pre",
                                "Afectivo_pre",
                                "Cognitivo_pre",
                                "Conativo_pre")),
                                by = "Código")

pivot_to_bind = dplyr::select(pre_post, c("Código", "Total_pre",
                                           "Afectivo_pre",
                                           "Cognitivo_pre",
                                           "Conativo_pre",
                                           "Total_post",
                                           "Afectivo_pre",
                                           "Cognitivo_post",
                                           "Conativo_post"))

pivot_to_bind[,2:ncol(pivot_to_bind)] =
  apply(pivot_to_bind[,2:ncol(pivot_to_bind)],
        2, function(x) min_max_scale(x))

pivot_to_bind =
  pivot_to_bind %>%
  pivot_longer(cols = !`Código`,
               names_to = c("Prueba", "Tipo"),
               names_sep = "_",
               values_to = "score")

pivot_to_bind$Prueba = str_replace_all(pivot_to_bind$Prueba,
                                         c("Total" = "Actitudes hacia las matematicas - Total",
                                             "Afectivo" = "Actitudes hacia las matematicas - Afectivo",
                                             "Cognitivo" = "Actitudes hacia las matematicas - Cognitivo",
                                             "Conativo" = "Actitudes hacia las matematicas - Conativo"))

pivot_final = rbind(pivot_final, pivot_to_bind)

```

3.1.2.3 Comparación pre-post Pese a que la mayoría de las pruebas conservan el mismo número de preguntas finales, no es el caso para todas. Es por eso que, con el fin de permitir la comparación entre las pruebas pre y post, haremos un escalamiento min-max tal que todas las puntuaciones queden entre 0 y 1.

```

pre_post$Total_pre = min_max_scale(pre_post$Total_pre)
pre_post$Total_post = min_max_scale(pre_post$Total_post)

```

3.1.2.3.1 Prueba total Estadísticos de normalidad

```

print("Estadístico de normalidad pre")

```

```
## [1] "Estadístico de normalidad pre"
lillie.test(pre_post$Total_pre)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_pre
## D = 0.291, p-value <0.0000000000000002
print("Estadístico de normalidad post")

## [1] "Estadístico de normalidad post"
lillie.test(pre_post$Total_post)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_post
## D = 0.365, p-value <0.0000000000000002
```

Debido a que no existe normalidad en las puntuaciones, se usará una prueba no paramétrica

Descriptivos

```
summ = data.frame(unclass(psych::describe(pre_post[,c("Total_pre",
                                                    "Total_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Total_pre", "Total_post")

summ[,1:10] %>%
  gt()
```

vars	n	mean	sd	median	trimmed	mad	min	max	range
Total_pre	452	0.88576	0.17724	1	0.92416	0	0	1	1
Total_post	452	0.91372	0.16993	1	0.95706	0	0	1	1

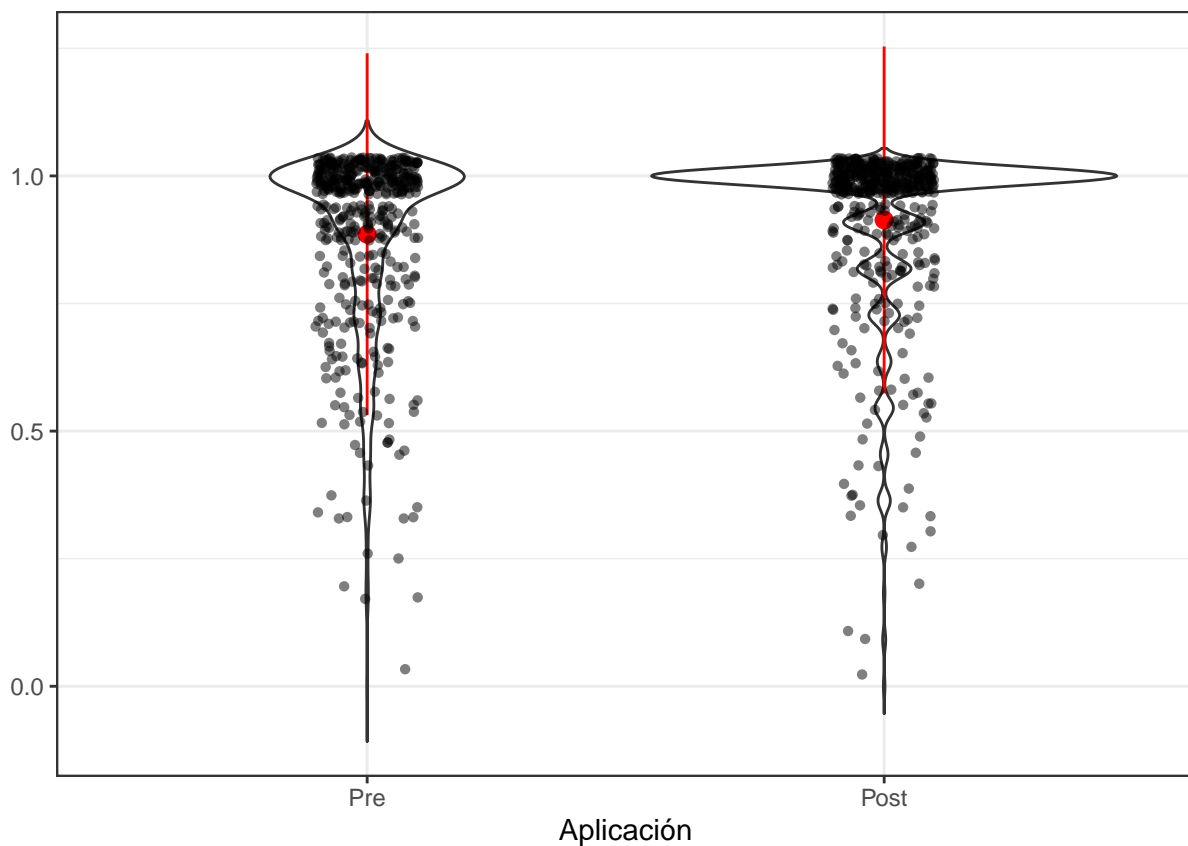
Comparación de medias

```
comparacion =
  wilcox.test(
    x = pre_post$Total_pre,
    y = pre_post$Total_post,
    alternative = "two.sided",
    mu = 0,
    var.equal = TRUE,
    paired = TRUE,
    conf.level = 0.95
  )
```

comparacion

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Total_pre and pre_post$Total_post
## V = 9414, p-value = 0.00048
## alternative hypothesis: true location shift is not equal to 0

pre_post %>%
  dplyr::select(Total_pre, Total_post) %>%
  pivot_longer(cols = c(Total_pre, Total_post),
               values_to = "value",
               names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
  scale_x_discrete(labels = c("Pre", "Post"))
```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/actitudes_matematicas.png")
```

Tamaño del efecto

```
size_effect =  
  cohen.d(pre_post$Total_post, pre_post$Total_pre, paired = TRUE)  
  
size_effect
```

```
##  
## Cohen's d  
##  
## d estimate: 0.16097 (negligible)  
## 95 percent confidence interval:  
##   lower   upper  
## 0.059684 0.262257
```

```
# Separamos los descriptivos importantes
```

```
desc = summ[,1:10] %>%  
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%  
  mutate(Prueba = "Actitudes hacia las matematicas - Total") %>%  
  mutate(Area = "Actitudes hacia las matematicas") %>%  
  mutate(`Aplicación` = c("Pre", "Post"))
```

```
descriptivos = rbind(descriptivos, desc)
```

```
# Guardamos los resultados de esta comparación
```

```
data_temp =  
  data.frame(Prueba = "Actitudes hacia las matematicas - Total",  
    `Media pre` = summ$mean[1],  
    `Media post` = summ$mean[2],  
    `p value` = comparacion$p.value,  
    `D de cohen` = size_effect$estimate)
```

```
comparaciones = rbind(comparaciones, data_temp)
```

```
pre_post$Afectivo_pre = min_max_scale(pre_post$Afectivo_pre)  
pre_post$Afectivo_post = min_max_scale(pre_post$Afectivo_post)
```

3.1.2.3.2 Componente afectivo Estadísticos de normalidad

```
print("Estadístico de normalidad pre")
```

```
## [1] "Estadístico de normalidad pre"
```



```

lillie.test(pre_post$Afectivo_pre)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Afectivo_pre
## D = 0.386, p-value <0.0000000000000002

print("Estadístico de normalidad post")

## [1] "Estadístico de normalidad post"

lillie.test(pre_post$Afectivo_post)

```

```

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Afectivo_post
## D = 0.471, p-value <0.0000000000000002

```

Debido a que no existe normalidad en las puntuaciones, se usará una prueba no paramétrica

Descriptivos

```

summ = data.frame(unclass(psych::describe(pre_post[,c("Afectivo_pre",
                                                       "Afectivo_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Afectivo_pre", "Afectivo_post")

summ[,1:10] %>%
  gt()

```

vars	n	mean	sd	median	trimmed	mad	min	max	range
Afectivo_pre	452	0.82743	0.26769	1	0.88122	0	0	1	1
Afectivo_post	452	0.90487	0.21069	1	0.95672	0	0	1	1

Comparación de medias

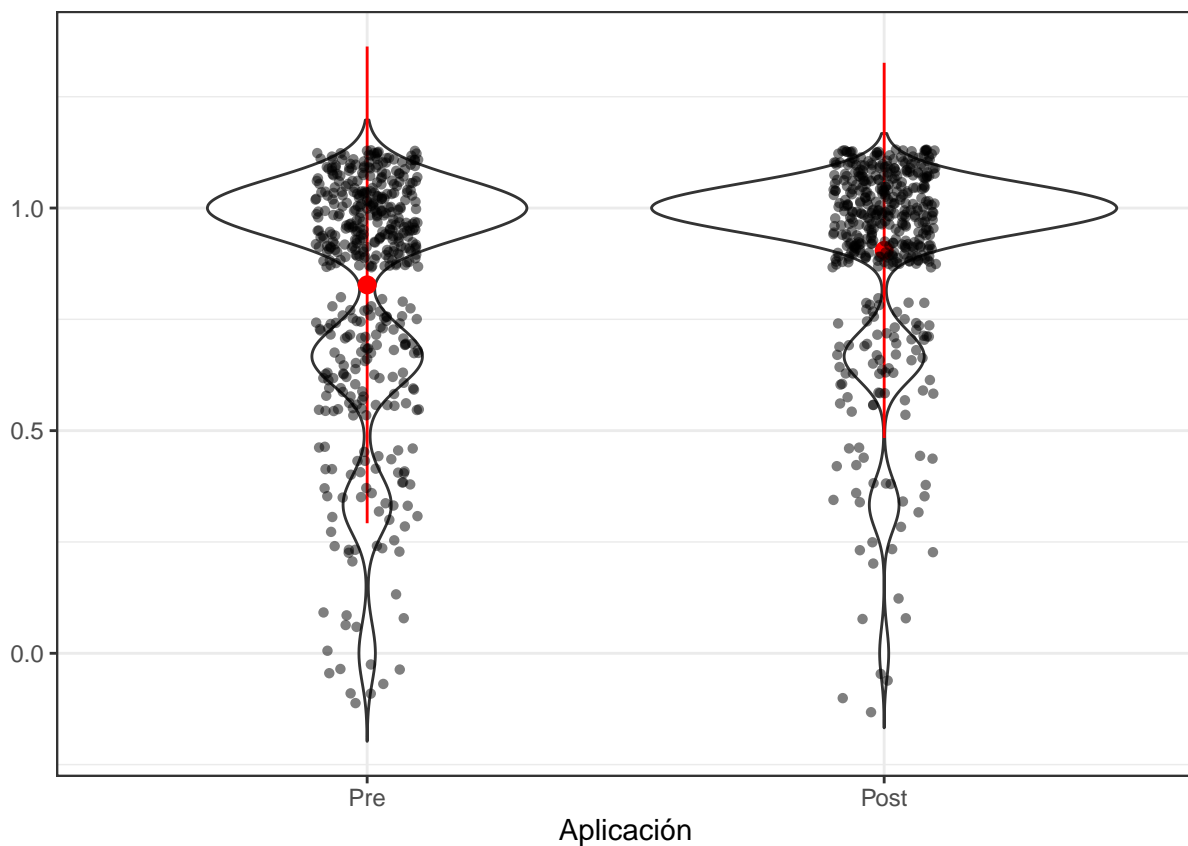
```

comparacion =
  wilcox.test(
    x      = pre_post$Afectivo_pre,
    y      = pre_post$Afectivo_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )

```

comparacion

```
##  
## Wilcoxon signed rank test with continuity correction  
##  
## data: pre_post$Afectivo_pre and pre_post$Afectivo_post  
## V = 4512, p-value = 0.00000069  
## alternative hypothesis: true location shift is not equal to 0  
  
pre_post %>%  
  dplyr::select(Afectivo_pre, Afectivo_post) %>%  
  pivot_longer(cols = c(Afectivo_pre, Afectivo_post),  
               values_to = "value",  
               names_to = "Aplicación") %>%  
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +  
  geom_violin(trim=FALSE) +  
  theme_bw() +  
  theme(legend.position = "none") +  
  xlab("Aplicación") + ylab("") +  
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +  
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +  
  scale_x_discrete(labels = c("Pre", "Post"))
```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/actitudes_matematicas_afectivo.png")
```

Tamaño del efecto

```
size_effect =  
  cohen.d(pre_post$Afectivo_post, pre_post$Afectivo_pre, paired = TRUE)  
  
size_effect
```

```
##  
## Cohen's d  
##  
## d estimate: 0.31993 (small)  
## 95 percent confidence interval:  
##   lower   upper  
## 0.20435 0.43552
```

```
# Separamos los descriptivos importantes
```

```
desc = summ[,1:10] %>%  
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%  
  mutate(Prueba = "Actitudes hacia las matematicas - Afectivo") %>%  
  mutate(Area = "Actitudes hacia las matematicas") %>%  
  mutate(`Aplicación` = c("Pre", "Post"))
```

```
descriptivos = rbind(descriptivos, desc)
```

```
# Guardamos los resultados de esta comparación
```

```
data_temp =  
  data.frame(Prueba = "Actitudes hacia las matematicas - Afectivo",  
    `Media pre` = summ$mean[1],  
    `Media post` = summ$mean[2],  
    `p value` = comparacion$p.value,  
    `D de cohen` = size_effect$estimate)
```

```
comparaciones = rbind(comparaciones, data_temp)
```

```
pre_post$Cognitivo_pre = min_max_scale(pre_post$Cognitivo_pre)  
pre_post$Cognitivo_post = min_max_scale(pre_post$Cognitivo_post)
```

3.1.2.3.3 Componente Cognitivo Estadísticos de normalidad

```
print("Estadístico de normalidad pre")
```

```
## [1] "Estadístico de normalidad pre"
```

```

lillie.test(pre_post$Cognitivo_pre)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Cognitivo_pre
## D = 0.451, p-value <0.0000000000000002

print("Estadístico de normalidad post")

## [1] "Estadístico de normalidad post"

lillie.test(pre_post$Cognitivo_post)

```

```

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Cognitivo_post
## D = 0.478, p-value <0.0000000000000002

```

Debido a que no existe normalidad en las puntuaciones, se usará una prueba no paramétrica

Descriptivos

```

summ = data.frame(unclass(psych::describe(pre_post[,c("Cognitivo_pre",
                                                       "Cognitivo_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Cognitivo_pre", "Cognitivo_post")

summ[,1:10] %>%
  gt()

```

vars	n	mean	sd	median	trimmed	mad	min	max	range
Cognitivo_pre	452	0.90819	0.19020	1	0.95787	0	0	1	1
Cognitivo_post	452	0.92865	0.18168	1	0.97652	0	0	1	1

Comparación de medias

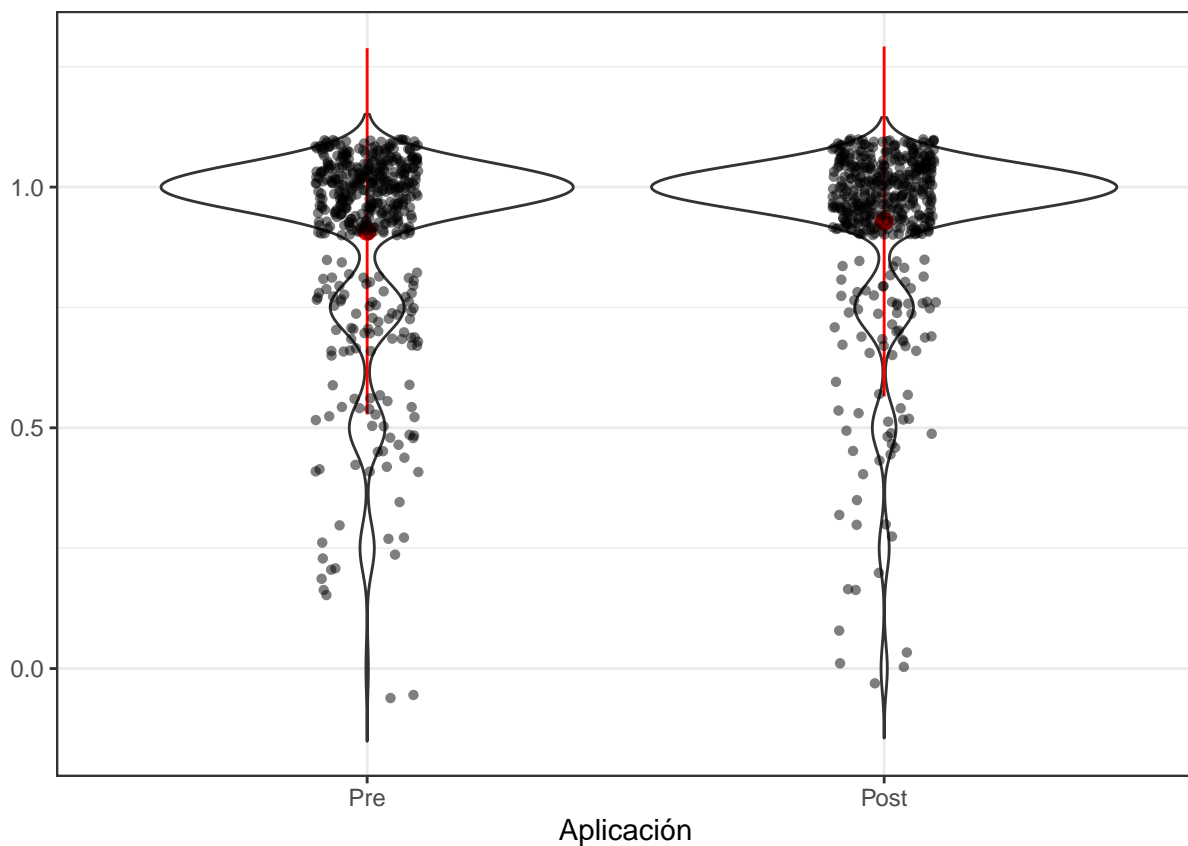
```

comparacion =
  wilcox.test(
    x      = pre_post$Cognitivo_pre,
    y      = pre_post$Cognitivo_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )

```

comparacion

```
##  
## Wilcoxon signed rank test with continuity correction  
##  
## data: pre_post$Cognitivo_pre and pre_post$Cognitivo_post  
## V = 3328, p-value = 0.035  
## alternative hypothesis: true location shift is not equal to 0  
  
pre_post %>%  
  dplyr::select(Cognitivo_pre, Cognitivo_post) %>%  
  pivot_longer(cols = c(Cognitivo_pre, Cognitivo_post),  
               values_to = "value",  
               names_to = "Aplicación") %>%  
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +  
  geom_violin(trim=FALSE) +  
  theme_bw() +  
  theme(legend.position = "none") +  
  xlab("Aplicación") + ylab("") +  
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +  
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +  
  scale_x_discrete(labels = c("Pre", "Post"))
```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/actitudes_matematicas_cognitivo.png")
```

Tamaño del efecto

```
size_effect =  
  cohen.d(pre_post$Cognitivo_post, pre_post$Cognitivo_pre, paired = TRUE)  
  
size_effect
```

```
##  
## Cohen's d  
##  
## d estimate: 0.11001 (negligible)  
## 95 percent confidence interval:  
##      lower      upper  
## -0.00085046  0.22086991
```

```
# Separamos los descriptivos importantes
```

```
desc = summ[,1:10] %>%  
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%  
  mutate(Prueba = "Actitudes hacia las matematicas - Cognitivo") %>%  
  mutate(Area = "Actitudes hacia las matematicas") %>%  
  mutate(`Aplicación` = c("Pre", "Post"))
```

```
descriptivos = rbind(descriptivos, desc)
```

```
# Guardamos los resultados de esta comparación
```

```
data_temp =  
  data.frame(Prueba = "Actitudes hacia las matematicas - Cognitivo",  
    `Media pre` = summ$mean[1],  
    `Media post` = summ$mean[2],  
    `p value` = comparacion$p.value,  
    `D de cohen` = size_effect$estimate)
```

```
comparaciones = rbind(comparaciones, data_temp)
```

```
pre_post$Conativo_pre = min_max_scale(pre_post$Conativo_pre)  
pre_post$Conativo_post = min_max_scale(pre_post$Conativo_post)
```

3.1.2.3.4 Componente Conativo Estadísticos de normalidad

```
print("Estadístico de normalidad pre")
```

```
## [1] "Estadístico de normalidad pre"
```

```

lillie.test(pre_post$Conativo_pre)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Conativo_pre
## D = 0.442, p-value <0.0000000000000002

print("Estadístico de normalidad post")

## [1] "Estadístico de normalidad post"

lillie.test(pre_post$Conativo_post)

```

```

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Conativo_post
## D = 0.451, p-value <0.0000000000000002

```

Debido a que no existe normalidad en las puntuaciones, se usará una prueba no paramétrica

Descriptivos

```

summ = data.frame(unclass(psych::describe(pre_post[,c("Conativo_pre",
                                                       "Conativo_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Conativo_pre", "Conativo_post")

summ[,1:10] %>%
  gt()

```

vars	n	mean	sd	median	trimmed	mad	min	max	range
Conativo_pre	452	0.90708	0.19039	1	0.95442	0	0	1	1
Conativo_post	452	0.90542	0.20058	1	0.95925	0	0	1	1

Comparación de medias

```

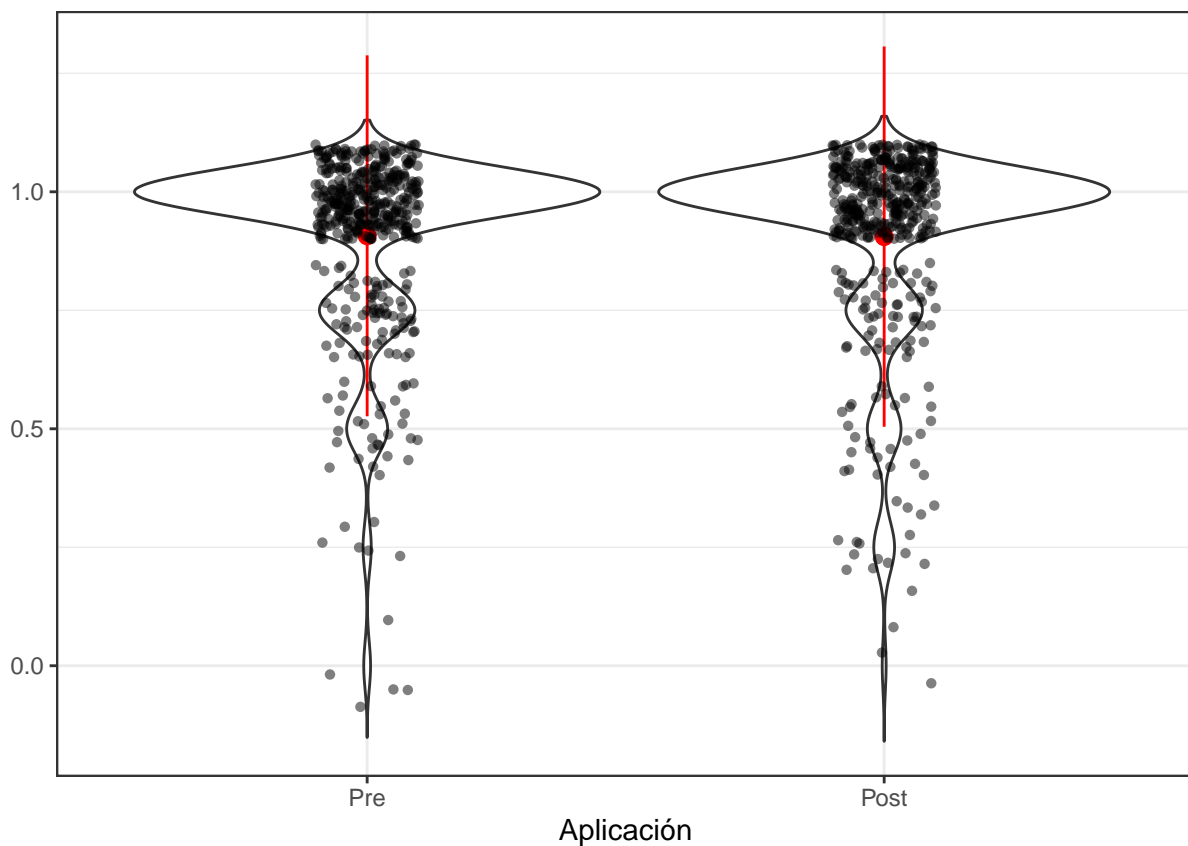
comparacion =
  wilcox.test(
    x      = pre_post$Conativo_pre,
    y      = pre_post$Conativo_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )

```

comparacion

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Conativo_pre and pre_post$Conativo_post
## V = 5404, p-value = 0.7
## alternative hypothesis: true location shift is not equal to 0

pre_post %>%
  dplyr::select(Conativo_pre, Conativo_post) %>%
  pivot_longer(cols = c(Conativo_pre, Conativo_post),
               values_to = "value",
               names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
  scale_x_discrete(labels = c("Pre", "Post"))
```




```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/actitudes_matematicas_conativo.png")
```

Tamaño del efecto

```
size_effect =  
  cohen.d(pre_post$Conativo_post, pre_post$Conativo_pre, paired = TRUE)  
  
size_effect
```

```
##  
## Cohen's d  
##  
## d estimate: -0.0084821 (negligible)  
## 95 percent confidence interval:  
##      lower      upper  
## -0.112852  0.095888
```

```
# Separamos los descriptivos importantes
```

```
desc = summ[,1:10] %>%  
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%  
  mutate(Prueba = "Actitudes hacia las matematicas - Conativo") %>%  
  mutate(Area = "Actitudes hacia las matematicas") %>%  
  mutate(`Aplicación` = c("Pre", "Post"))
```

```
descriptivos = rbind(descriptivos, desc)
```

```
# Guardamos los resultados de esta comparación
```

```
data_temp =  
  data.frame(Prueba = "Actitudes hacia las matematicas - Conativo",  
    `Media pre` = summ$mean[1],  
    `Media post` = summ$mean[2],  
    `p value` = comparacion$p.value,  
    `D de cohen` = size_effect$estimate)
```

```
comparaciones = rbind(comparaciones, data_temp)
```

3.2 Motivación

```
# Motivación Pre
```

```
motivacion_pre = read_sheet(url_pre, sheet = "Motivación")  
motivacion_pre = motivacion_pre[,1:16]
```

```
motivacion_pre = filter(motivacion_pre,  
  !is.na(`Código`),  
  !is.na(`1`))
```

```

motivacion_pre[,7:16] = apply(motivacion_pre[,7:16], 2,
                             function(x) str_to_upper(x))

claves_motivacion_pre = c('B','B','A','B','A','A','B','B','A','A')

motivacion_pre[,7:16] = calificacion(motivacion_pre[,7:16],
                                     claves_motivacion_pre)

# Motivación Post

motivacion_post = read_sheet(url_post, sheet = "Motivación")
motivacion_post = motivacion_post[,1:16]

motivacion_post = filter(motivacion_post,
                         !is.na(`Código`),
                         !is.na(`1`))

motivacion_post[,7:16] = apply(motivacion_post[,7:16], 2,
                              function(x) str_to_upper(x))

claves_motivacion_post = c('B','A','A','B','B','B','B','B','A','A')

motivacion_post[,7:16] = calificacion(motivacion_post[,7:16],
                                     claves_motivacion_post)

# Calificación

# General

motivacion_pre$Total_pre = apply(motivacion_pre[,7:16], 1,
                                function(x) sum(x, na.rm= FALSE))

motivacion_post$Total_post = apply(motivacion_post[,7:16], 1,
                                   function(x) sum(x, na.rm= FALSE))

# Interés

motivacion_pre$Interes_pre = apply(motivacion_pre[,c(8,11,14)], 1,
                                   function(x) sum(x, na.rm= FALSE))

motivacion_post$Interes_post = apply(motivacion_post[,c(7,10,13)], 1,
                                     function(x) sum(x, na.rm= FALSE))

## Metas

```

```

# Orientación al aprendizaje

motivacion_pre$OrientacionAprendizaje_pre =
  apply(motivacion_pre[,c(9,12,15)], 1,
        function(x) sum(x, na.rm= FALSE))

motivacion_post$OrientacionAprendizaje_post =
  apply(motivacion_post[,c(8,11,15)], 1,
        function(x) sum(x, na.rm= FALSE))

# Orientación al resultado

motivacion_pre$OrientacionResultado_pre =
  3 - motivacion_pre$OrientacionAprendizaje_pre

motivacion_post$OrientacionResultado_post =
  3 - motivacion_post$OrientacionAprendizaje_post

# Atribución interna

motivacion_pre$AtribucionInterna_pre =
  apply(motivacion_pre[,c(13,16)], 1,
        function(x) sum(x, na.rm= FALSE))

motivacion_post$AtribucionInterna_post =
  apply(motivacion_post[,c(9,16)], 1,
        function(x) sum(x, na.rm= FALSE))

# Atribución externa

motivacion_pre$AtribucionExterna_pre =
  2 - motivacion_pre$AtribucionInterna_pre

motivacion_post$AtribucionExterna_post =
  2 - motivacion_post$AtribucionInterna_post

# Expectativa

motivacion_pre$Expectativa_pre = apply(motivacion_pre[,c(7, 10)], 1,
                                         function(x) sum(x, na.rm= FALSE))

motivacion_post$Expectativa_post = apply(motivacion_post[,c(12, 14)], 1,
                                           function(x) sum(x, na.rm= FALSE))

columnas_pre = c("Total_pre",
                  "Interes_pre",
                  "OrientacionResultado_pre",

```

```

        "OrientacionAprendizaje_pre",
        "AtribucionInterna_pre",
        "AtribucionExterna_pre",
        "Expectativa_pre")

columnas_post = c("Total_post",
                  "Interes_post",
                  "OrientacionAprendizaje_post",
                  "OrientacionResultado_post",
                  "AtribucionInterna_post",
                  "AtribucionExterna_post",
                  "Expectativa_post")

motivacion_pre[,columnas_pre] = apply(motivacion_pre[,columnas_pre],2,
                                     function(x) min_max_scale(x) )

motivacion_post[,columnas_post] = apply(motivacion_post[,columnas_post],2,
                                     function(x) min_max_scale(x) )

pre_post = inner_join(motivacion_post,
                      dplyr::select(motivacion_pre, c("Código",
                                                         columnas_pre)),
                      by = "Código")

pivot_to_bind = dplyr::select(pre_post, c("Código",
                                           columnas_pre,
                                           columnas_post))

pivot_to_bind[,2:ncol(pivot_to_bind)] =
  apply(pivot_to_bind[,2:ncol(pivot_to_bind)],
        2, function(x) 1 - min_max_scale(x))

pivot_to_bind =
  pivot_to_bind %>%
  pivot_longer(cols = !`Código`,
              names_to = c("Prueba", "Tipo"),
              names_sep = "_",
              values_to = "score")

pivot_to_bind$Prueba = str_replace_all(pivot_to_bind$Prueba,
                                       c("Total" = "Motivacion - Total",
                                         "Interes" = "Motivacion - Interes",
                                         "OrientacionAprendizaje" = "Motivacion - Metas",
                                         "AtribucionInterna" = "Motivacion - Atribucion interna",
                                         "Expectativa" = "Motivacion - Expectativas"))

pivot_final = rbind(pivot_final, pivot_to_bind)

```

3.2.1 Alpha

```
alpha(motivacion_pre[,7:16])
```

3.2.1.1 Total pre

```
## [1] 0.29199
```

```
for(i in seq(length(colnames(motivacion_pre[,7:16])))){  
  x = alpha(motivacion_pre[,7:16][,-i])  
  print(paste("El índice de confiabilidad cambia a", x,  
             "al eliminar el ítem",  
             colnames(motivacion_pre[,7:16])[i]))  
}
```

```
## [1] "El índice de confiabilidad cambia a 0.229438611029352 al eliminar el ítem 1"  
## [1] "El índice de confiabilidad cambia a 0.177949167764277 al eliminar el ítem 2"  
## [1] "El índice de confiabilidad cambia a 0.199564804719596 al eliminar el ítem 3"  
## [1] "El índice de confiabilidad cambia a 0.202052882284707 al eliminar el ítem 4"  
## [1] "El índice de confiabilidad cambia a 0.201127531319761 al eliminar el ítem 5"  
## [1] "El índice de confiabilidad cambia a 0.231261726724843 al eliminar el ítem 6"  
## [1] "El índice de confiabilidad cambia a 0.250273703427014 al eliminar el ítem 7"  
## [1] "El índice de confiabilidad cambia a 0.35083992022258 al eliminar el ítem 8"  
## [1] "El índice de confiabilidad cambia a 0.45241665467227 al eliminar el ítem 9"  
## [1] "El índice de confiabilidad cambia a 0.333053418467337 al eliminar el ítem 10"
```

```
alpha(motivacion_post[,7:16])
```

3.2.1.2 Total post

```
## [1] 0.65516
```

```
for(i in seq(length(colnames(motivacion_post[,7:16])))){  
  x = alpha(motivacion_post[,7:16][,-i])  
  print(paste("El índice de confiabilidad cambia a", x,  
             "al eliminar el ítem",  
             colnames(motivacion_post[,7:16])[i]))  
}
```

```
## [1] "El índice de confiabilidad cambia a 0.615085704752272 al eliminar el ítem 1"  
## [1] "El índice de confiabilidad cambia a 0.612434574710164 al eliminar el ítem 2"  
## [1] "El índice de confiabilidad cambia a 0.645816237592549 al eliminar el ítem 3"  
## [1] "El índice de confiabilidad cambia a 0.614221838519653 al eliminar el ítem 4"  
## [1] "El índice de confiabilidad cambia a 0.632932803744745 al eliminar el ítem 5"  
## [1] "El índice de confiabilidad cambia a 0.621971077288762 al eliminar el ítem 6"  
## [1] "El índice de confiabilidad cambia a 0.611816787506972 al eliminar el ítem 7"  
## [1] "El índice de confiabilidad cambia a 0.6154170110309 al eliminar el ítem 8"  
## [1] "El índice de confiabilidad cambia a 0.662907727963067 al eliminar el ítem 9"  
## [1] "El índice de confiabilidad cambia a 0.673778504431847 al eliminar el ítem 10"
```

Si bien el alfa del pre es bajo, en el post este sube

```
# Amacenos los valores
```

```
alfa = rbind(alfa,  
             c("Motivación pre",  
               alpha(motivacion_pre[,7:16])))  
  
alfa = rbind(alfa,  
             c("Motivación post",  
               alpha(motivacion_post[,7:16])))
```

```
alpha(motivacion_pre[,c(8,11,14)])
```

3.2.1.3 Interés pre

```
## [1] 0.036188
```

```
for(i in seq(length(colnames(motivacion_pre[,c(8,11,14)])))){  
  x = alpha(motivacion_pre[,c(8,11,14)][,-i])  
  print(paste("El índice de confiabilidad cambia a", x,  
              "al eliminar el ítem",  
              colnames(motivacion_pre[,c(8,11,14)])[i]))  
}
```

```
## [1] "El índice de confiabilidad cambia a -0.491432198011423 al eliminar el ítem 2"  
## [1] "El índice de confiabilidad cambia a -0.217013313118871 al eliminar el ítem 5"  
## [1] "El índice de confiabilidad cambia a 0.411888027403296 al eliminar el ítem 8"
```

```
alpha(motivacion_post[,c(7,10,13)])
```

3.2.1.4 Interés post

```
## [1] 0.63595
```

```
for(i in seq(length(colnames(motivacion_post[,c(7,10,13)])))){  
  x = alpha(motivacion_post[,c(7,10,13)][,-i])  
  print(paste("El índice de confiabilidad cambia a", x,  
              "al eliminar el ítem",  
              colnames(motivacion_post[,c(7,10,13)])[i]))  
}
```

```
## [1] "El índice de confiabilidad cambia a 0.539037134445078 al eliminar el ítem 1"  
## [1] "El índice de confiabilidad cambia a 0.506862111915856 al eliminar el ítem 4"  
## [1] "El índice de confiabilidad cambia a 0.566275670915211 al eliminar el ítem 7"
```

```
alpha(motivacion_pre[,c(9,12,15)])
```

3.2.1.5 Metas pre

```
## [1] -0.24489
```

```
for(i in seq(length(colnames(motivacion_pre[,c(9,12,15)])))){  
  x = alpha(motivacion_pre[,c(9,12,15)][,-i])  
  print(paste("El índice de confiabilidad cambia a", x,  
              "al eliminar el ítem",  
              colnames(motivacion_pre[,c(9,12,15)])[i]))  
}
```

```
## [1] "El índice de confiabilidad cambia a -0.528262612003685 al eliminar el ítem 3"
```

```
## [1] "El índice de confiabilidad cambia a -0.391812763685533 al eliminar el ítem 6"
```

```
## [1] "El índice de confiabilidad cambia a 0.318044239573478 al eliminar el ítem 9"
```

```
alpha(motivacion_post[,c(8,11,15)])
```

3.2.1.6 Metas post

```
## [1] 0.37252
```

```
for(i in seq(length(colnames(motivacion_post[,c(8,11,15)])))){  
  x = alpha(motivacion_post[,c(8,11,15)][,-i])  
  print(paste("El índice de confiabilidad cambia a", x,  
              "al eliminar el ítem",  
              colnames(motivacion_post[,c(8,11,15)])[i]))  
}
```

```
## [1] "El índice de confiabilidad cambia a 0.206854665051957 al eliminar el ítem 2"
```

```
## [1] "El índice de confiabilidad cambia a 0.394686747843145 al eliminar el ítem 5"
```

```
## [1] "El índice de confiabilidad cambia a 0.226526583997217 al eliminar el ítem 9"
```

```
alpha(motivacion_pre[,c(13,16)])
```

3.2.1.7 Atribución interna pre

```
## [1] -0.17208
```

```
alpha(motivacion_post[,c(9,16)])
```

3.2.1.8 Atribución interna pre

```
## [1] 0.11387
```

No puede obtener el indicador al eliminar un ítem, ya que si queda un único ítem el indicador no tiene solución.

```
alpha(motivacion_pre[,c(7,10)])
```

3.2.1.9 Expectativa positiva pre

```
## [1] 0.33053
```

```
alpha(motivacion_post[,c(12,14)])
```

3.2.1.10 Expectativa positiva post

```
## [1] 0.49718
```

3.2.2 Indicadores psicométricos

```
analitem = item.exam(motivacion_pre[,7:16], discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body()
```

3.2.2.1 Total pre

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.334	0.425	0.199	0.873	0.288	0.142
0.315	0.522	0.326	0.889	0.275	0.164
0.383	0.494	0.242	0.822	0.391	0.189
0.354	0.482	0.250	0.853	0.315	0.171
0.285	0.477	0.295	0.911	0.221	0.136
0.394	0.448	0.180	0.808	0.383	0.177
0.468	0.464	0.140	0.676	0.553	0.217
0.228	-0.015	-0.177	0.055	0.005	-0.003
0.430	0.041	-0.261	0.245	0.013	0.018

0.484	0.356	0.007	0.626	0.450	0.172
-------	-------	-------	-------	-------	-------

```
# Guardamos los indicadores importantes
```

```
psicometricos_pre = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Motivación - Pre") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_pre)
```

```
analitem = item.exam(motivacion_post[,7:16], discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body()
```

3.2.2.2 Total post

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.345	0.551	0.413	0.862	0.328	0.190
0.435	0.576	0.401	0.747	0.555	0.250
0.437	0.450	0.251	0.743	0.437	0.197
0.335	0.555	0.422	0.871	0.311	0.186
0.424	0.495	0.309	0.765	0.434	0.210
0.354	0.519	0.371	0.853	0.331	0.184
0.327	0.568	0.441	0.879	0.311	0.185
0.366	0.551	0.402	0.840	0.367	0.202

0.495	0.428	0.195	0.572	0.569	0.212
0.481	0.373	0.141	0.639	0.412	0.179

```
# Guardamos los indicadores importantes

psicometricos_pre = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Motivación - Post") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_pre)
```

3.2.3 Comparación pre-post

3.2.3.1 Prueba total

```
print("Estadístico de normalidad pre")
```

3.2.3.1.1 Estadísticos de normalidad

```
## [1] "Estadístico de normalidad pre"
```

```
lillie.test(pre_post$Total_pre)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_pre
## D = 0.232, p-value <0.0000000000000002
```

```
print("Estadístico de normalidad post")
```

```
## [1] "Estadístico de normalidad post"
```

```
lillie.test(pre_post$Total_post)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_post
## D = 0.198, p-value <0.0000000000000002
```

Debido a que no existe normalidad en las puntuaciones, se usará una prueba no paramétrica

```
summ = data.frame(unclass(psych::describe(pre_post[,c("Total_pre",
                                                       "Total_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)
```

```
summ$vars = c("Total_pre", "Total_post")

summ[,1:10] %>%
  gt()
```

3.2.3.1.2 Descriptivos

vars	n	mean	sd	median	trimmed	mad	min	max	range
Total_pre	887	0.67610	0.14195	0.7	0.69353	0.14826	0	1	1
Total_post	887	0.78805	0.19431	0.8	0.81505	0.14826	0	1	1

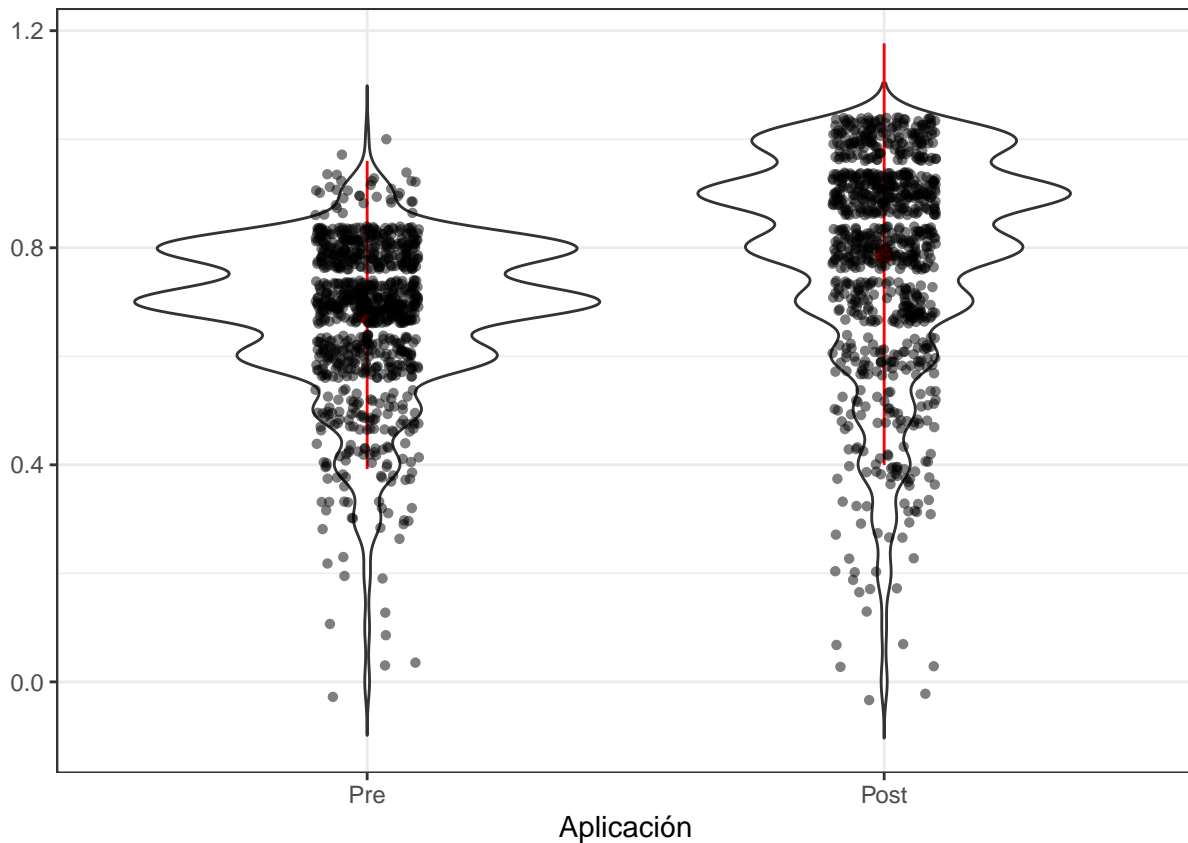
```
comparacion =
  wilcox.test(
    x      = pre_post$Total_pre,
    y      = pre_post$Total_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )

comparacion
```

3.2.3.1.3 Comparación de medias

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Total_pre and pre_post$Total_post
## V = 55084, p-value <0.0000000000000002
## alternative hypothesis: true location shift is not equal to 0

pre_post %>%
  dplyr::select(Total_pre, Total_post) %>%
  pivot_longer(cols = c(Total_pre, Total_post),
               values_to = "value",
               names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
  scale_x_discrete(labels = c("Pre", "Post"))
```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/motivacion.png")
```

```
size_effect =
  cohen.d(pre_post$Total_post, pre_post$Total_pre, paired = TRUE)

size_effect
```

3.2.3.1.4 Tamaño del efecto

```
##
## Cohen's d
##
## d estimate: 0.65224 (medium)
## 95 percent confidence interval:
## lower upper
## 0.56459 0.73990
```

```
# Separamos los descriptivos importantes
```

```
desc = summ[,1:10] %>%
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
  mutate(Prueba = "Motivación - Total") %>%
```

```

mutate(Area = "Motivación") %>%
mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Motivación - Total",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)

```

3.2.3.2 Interés

```
print("Estadístico de normalidad pre")
```

3.2.3.2.1 Estadísticos de normalidad

```
## [1] "Estadístico de normalidad pre"
```

```
lillie.test(pre_post$Interes_pre)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Interes_pre
## D = 0.469, p-value <0.0000000000000002
```

```
print("Estadístico de normalidad post")
```

```
## [1] "Estadístico de normalidad post"
```

```
lillie.test(pre_post$Interes_post)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Interes_post
## D = 0.453, p-value <0.0000000000000002
```

Debido a que no existe normalidad en las puntuaciones, se usará una prueba no paramétrica

```
summ = data.frame(unclass(psych::describe(pre_post[,c("Interes_pre",
  "Interes_post")])),
```

```

        check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Interes_pre", "Interes_post")

summ[,1:10] %>%
  gt()

```

3.2.3.2.2 Descriptivos

vars	n	mean	sd	median	trimmed	mad	min	max	range
Interes_pre	887	0.61969	0.16127	0.66667	0.64744	0	0	1	1
Interes_post	887	0.87862	0.24975	1.00000	0.94421	0	0	1	1

```

comparacion =
  wilcox.test(
    x      = pre_post$Interes_pre,
    y      = pre_post$Interes_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )

comparacion

```

3.2.3.2.3 Comparación de medias

```

##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Interes_pre and pre_post$Interes_post
## V = 10734, p-value <0.0000000000000002
## alternative hypothesis: true location shift is not equal to 0

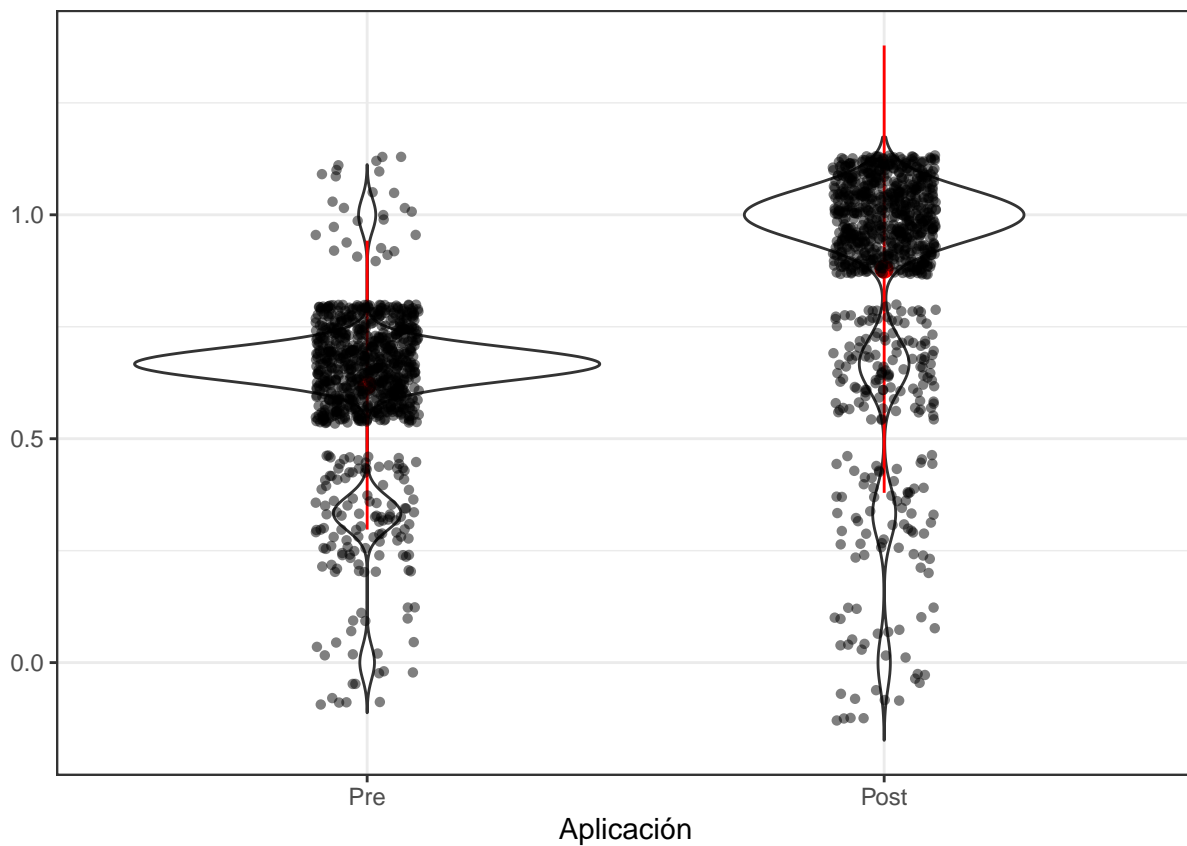
```

```

pre_post %>%
  dplyr::select(Interes_pre, Interes_post) %>%
  pivot_longer(cols = c(Interes_pre, Interes_post),
    values_to = "value",
    names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +

```

```
geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
scale_x_discrete(labels = c("Pre", "Post"))
```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/motivacion_interes.png")
```

```
size_effect =
  cohen.d(pre_post$Interes_post, pre_post$Interes_pre, paired = TRUE)

size_effect
```

3.2.3.2.4 Tamaño del efecto

```
##
## Cohen's d
##
## d estimate: 1.2118 (large)
## 95 percent confidence interval:
## lower upper
## 1.1072 1.3164
```

```

# Separamos los descriptivos importantes
desc = summ[,1:10] %>%
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
  mutate(Prueba = "Motivación - Interés") %>%
  mutate(Area = "Motivación") %>%
  mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Motivación - Interés",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)

```

3.2.3.3 Metas Orientación al aprendizaje

```
print("Estadístico de normalidad pre")
```

3.2.3.3.1 Estadísticos de normalidad

```
## [1] "Estadístico de normalidad pre"
```

```
lillie.test(pre_post$OrientacionAprendizaje_pre)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$OrientacionAprendizaje_pre
## D = 0.365, p-value <0.0000000000000002
```

```
print("Estadístico de normalidad post")
```

```
## [1] "Estadístico de normalidad post"
```

```
lillie.test(pre_post$OrientacionAprendizaje_post)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$OrientacionAprendizaje_post
## D = 0.249, p-value <0.0000000000000002
```

Debido a que no existe normalidad en las puntuaciones, se usará una prueba no paramétrica


```
summ = data.frame(unclass(psych::describe(pre_post[,c("OrientacionAprendizaje_pre",
                                                    "OrientacionAprendizaje_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("OrientacionAprendizaje_pre", "OrientacionAprendizaje_post")

summ[,1:10] %>%
  gt()
```

3.2.3.3.2 Descriptivos

vars	n	mean	sd	median	trimmed	mad	min	max	range
OrientacionAprendizaje_pre	887	0.6242	0.21962	0.66667	0.62963	0.0000	0	1	1
OrientacionAprendizaje_post	887	0.7050	0.29929	0.66667	0.73746	0.4942	0	1	1

```
comparacion =
  wilcox.test(
    x      = pre_post$OrientacionAprendizaje_pre,
    y      = pre_post$OrientacionAprendizaje_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )
```

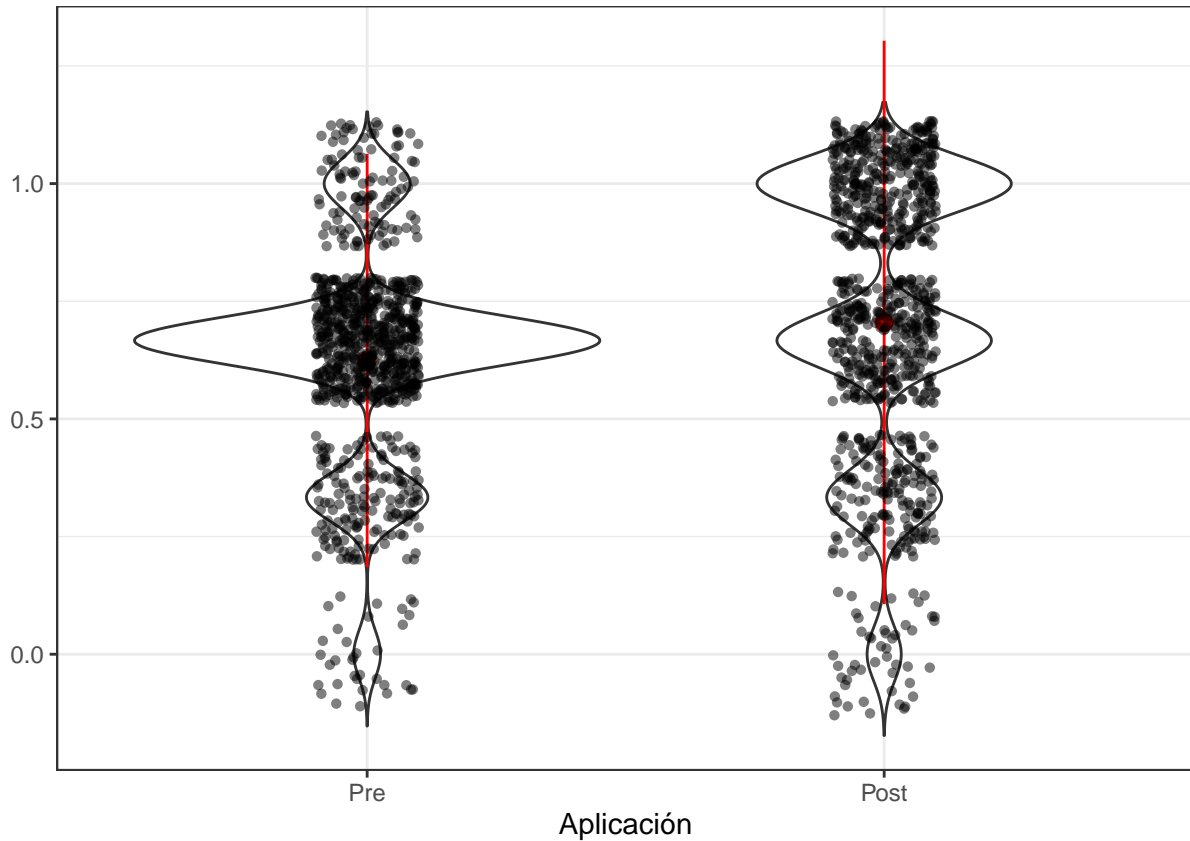
```
comparacion
```

3.2.3.3.3 Comparación de medias

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$OrientacionAprendizaje_pre and pre_post$OrientacionAprendizaje_post
## V = 53251, p-value <0.0000000000000002
## alternative hypothesis: true location shift is not equal to 0
```

```
pre_post %>%
  dplyr::select(OrientacionAprendizaje_pre, OrientacionAprendizaje_post) %>%
  pivot_longer(cols = c(OrientacionAprendizaje_pre, OrientacionAprendizaje_post),
               values_to = "value",
               names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
  theme(legend.position = "none") +
```

```
xlab("Aplicación") + ylab("") +
stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +
geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
scale_x_discrete(labels = c("Pre", "Post"))
```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/motivacion_metas_aprendizaje.png")
```

```
size_effect =
  cohen.d(pre_post$OrientacionAprendizaje_post,
          pre_post$OrientacionAprendizaje_pre, paired = TRUE)

size_effect
```

3.2.3.3.4 Tamaño del efecto

```
##
## Cohen's d
##
## d estimate: 0.30706 (small)
## 95 percent confidence interval:
```

```
## lower upper
## 0.21642 0.39770

# Separamos los descriptivos importantes
desc = summ[,1:10] %>%
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
  mutate(Prueba = "Motivación - Metas") %>%
  mutate(Area = "Motivación") %>%
  mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Motivación - Metas",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)
```

3.2.3.4 Atribución interna

```
print("Estadístico de normalidad pre")
```

3.2.3.4.1 Estadísticos de normalidad

```
## [1] "Estadístico de normalidad pre"
lillie.test(pre_post$AtribucionInterna_pre)

##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: pre_post$AtribucionInterna_pre
## D = 0.275, p-value <0.0000000000000002
print("Estadístico de normalidad post")

## [1] "Estadístico de normalidad post"
lillie.test(pre_post$AtribucionInterna_post)

##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: pre_post$AtribucionInterna_post
## D = 0.323, p-value <0.0000000000000002
```

Debido a que no existe normalidad en las puntuaciones, se usará una prueba no paramétrica

```
summ = data.frame(unclass(psych::describe(pre_post[,c("AtribucionInterna_pre",
                                                    "AtribucionInterna_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("AtribucionInterna_pre", "AtribucionInterna_post")

summ[,1:10] %>%
  gt()
```

3.2.3.4.2 Descriptivos

vars	n	mean	sd	median	trimmed	mad	min	max	range
AtribucionInterna_pre	887	0.65614	0.32158	0.5	0.69480	0.7413	0	1	1
AtribucionInterna_post	887	0.70857	0.32730	1.0	0.75598	0.0000	0	1	1

```
comparacion =
  wilcox.test(
    x      = pre_post$AtribucionInterna_pre,
    y      = pre_post$AtribucionInterna_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )

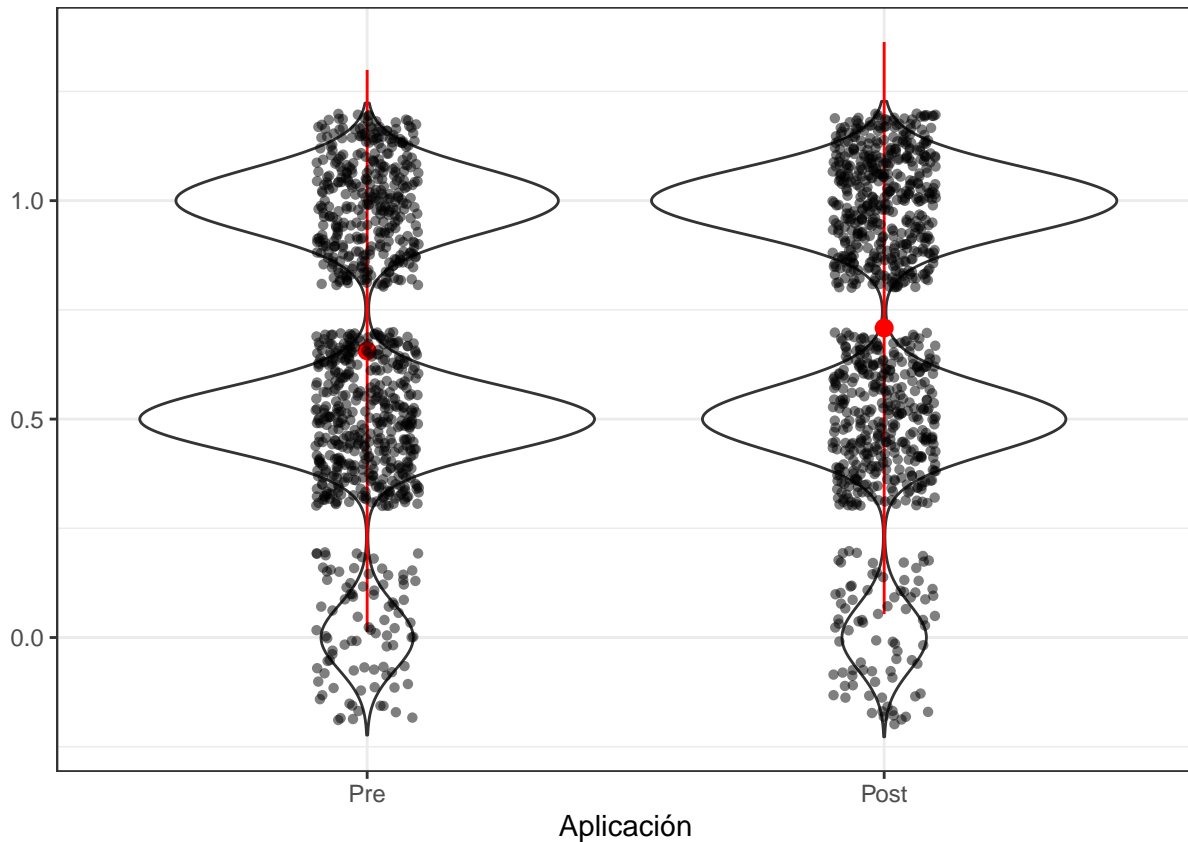
comparacion
```

3.2.3.4.3 Comparación de medias

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$AtribucionInterna_pre and pre_post$AtribucionInterna_post
## V = 45575, p-value = 0.00028
## alternative hypothesis: true location shift is not equal to 0

pre_post %>%
  dplyr::select(AtribucionInterna_pre, AtribucionInterna_post) %>%
  pivot_longer(cols = c(AtribucionInterna_pre, AtribucionInterna_post),
               values_to = "value",
               names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
```

```
theme_bw() +
  theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
  scale_x_discrete(labels = c("Pre", "Post"))
```



Guardamos la imagen

```
ggsave("../Plots/segunda_aplicacion/motivacion_AtribucionInterna.png")
```

```
size_effect =
  cohen.d(pre_post$AtribucionInterna_post,
          pre_post$AtribucionInterna_pre, paired = TRUE)

size_effect
```

3.2.3.4.4 Tamaño del efecto

```
##
## Cohen's d
##
```

```
## d estimate: 0.16157 (negligible)
## 95 percent confidence interval:
##      lower      upper
## 0.074948 0.248200

# Separamos los descriptivos importantes
desc = summ[,1:10] %>%
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
  mutate(Prueba = "Motivación - Atribución interna") %>%
  mutate(Area = "Motivación") %>%
  mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Motivación - Atribución interna",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)
```

3.2.3.5 Expectativas positivas

```
print("Estadístico de normalidad pre")
```

3.2.3.5.1 Estadísticos de normalidad

```
## [1] "Estadístico de normalidad pre"
```

```
lillie.test(pre_post$Expectativa_pre)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Expectativa_pre
## D = 0.461, p-value <0.0000000000000002
```

```
print("Estadístico de normalidad post")
```

```
## [1] "Estadístico de normalidad post"
```

```
lillie.test(pre_post$Expectativa_post)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
```

```
## data: pre_post$Expectativa_post
## D = 0.466, p-value <0.0000000000000002
```

Debido a que no existe normalidad en las puntuaciones, se usará una prueba no paramétrica

```
summ = data.frame(unclass(psych::describe(pre_post[,c("Expectativa_pre",
                                                       "Expectativa_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Expectativa_pre", "Expectativa_post")

summ[,1:10] %>%
  gt()
```

3.2.3.5.2 Descriptivos

vars	n	mean	sd	median	trimmed	mad	min	max	range
Expectativa_pre	887	0.85851	0.2698	1	0.91280	0	0	1	1
Expectativa_post	887	0.85626	0.2859	1	0.92053	0	0	1	1

```
comparacion =
  wilcox.test(
    x = pre_post$Expectativa_pre,
    y = pre_post$Expectativa_post,
    alternative = "two.sided",
    mu = 0,
    var.equal = TRUE,
    paired = TRUE,
    conf.level = 0.95
  )

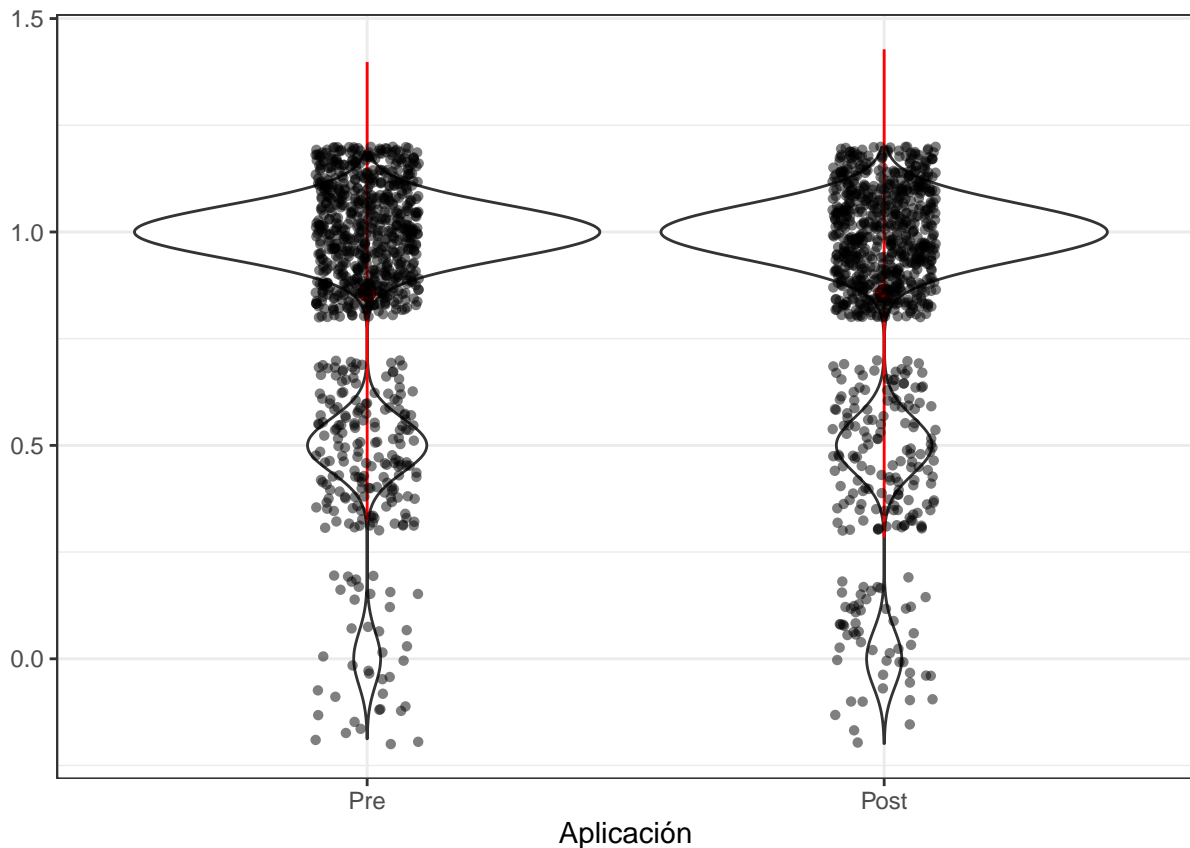
comparacion
```

3.2.3.5.3 Comparación de medias

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Expectativa_pre and pre_post$Expectativa_post
## V = 20015, p-value = 0.78
## alternative hypothesis: true location shift is not equal to 0

pre_post %>%
  dplyr::select(Expectativa_pre, Expectativa_post) %>%
  pivot_longer(cols = c(Expectativa_pre, Expectativa_post),
               values_to = "value",
               names_to = "Aplicación") %>%
```

```
ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
  scale_x_discrete(labels = c("Pre", "Post"))
```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/motivacion_expectativas.png")
```

```
size_effect =
  cohen.d(pre_post$Expectativa_post,
          pre_post$Expectativa_pre, paired = TRUE)

size_effect
```

3.2.3.5.4 Tamaño del efecto

```
##
```



```
## Cohen's d
##
## d estimate: -0.0081094 (negligible)
## 95 percent confidence interval:
##      lower      upper
## -0.088864  0.072645

# Separamos los descriptivos importantes
desc = summ[,1:10] %>%
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
  mutate(Prueba = "Motivación - Expectativas") %>%
  mutate(Area = "Motivación") %>%
  mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Motivación - Expectativas",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)
```

3.3 Funciones ejecutivas

Para funciones ejecutivas no se calcula un puntaje global, ya que cada test apunta a evaluar un dominio distinto.

3.3.1 Memoria auditiva

```
# Pre

mem_audi_pre = read_sheet(url_pre, sheet = "Memoria Audi")
vector = c(colnames(mem_audi_pre)[1:6], as.vector(unlist(mem_audi_pre[1,][7:30])))
colnames(mem_audi_pre) = vector
mem_audi_pre = mem_audi_pre[-1,1:length(vector)]
mem_audi_pre = filter(mem_audi_pre,
  !is.na(`Código`),
  !is.na(Caballo),
  Caballo != 'NULL')

item_1_pre = c("Caballo", "Perro", "Aguila", "Pollito",
  "Foca", "Vaca", "Cocodrilo", "Sapo")

item_2_pre = c("Oveja", "Elefante", "Tiburon", "Caracol",
```

```

        "Ratón" , "Gato", "Tortuga", "Pez")

incorrectos_1_pre = c("León",    "Oso", "Ballena",    "Lagarto")
incorrectos_2_pre = c("Mariposa", "Mono",    "Jirafa",    "Araña")

mem_audi_pre[,7:30] = apply(mem_audi_pre[,7:30], 2,
                            function(x) str_replace_all(x,'NULL', '0'))

mem_audi_pre$`Aciertos item 1_pre` =
  apply(apply(mem_audi_pre[,item_1_pre], 2,
              function(x) as.numeric(x)),
        1, function(x) sum(x, na.rm= FALSE))

mem_audi_pre$`Errores item 1_pre` =
  apply(apply(mem_audi_pre[,incorrectos_1_pre],
              2, function(x) as.numeric(x)),
        1, function(x) sum(x, na.rm= FALSE))

mem_audi_pre$`Aciertos item 2_pre` =
  apply(apply(mem_audi_pre[,item_2_pre], 2,
              function(x) as.numeric(x)),
        1, function(x) sum(x, na.rm= FALSE))

mem_audi_pre$`Errores item 2_pre` =
  apply(apply(mem_audi_pre[,incorrectos_2_pre],
              2, function(x) as.numeric(x)),
        1, function(x) sum(x, na.rm= FALSE))

mem_audi_pre$`Puntaje item 1_pre` =
  mem_audi_pre$`Aciertos item 1_pre` - mem_audi_pre$`Errores item 1_pre`

mem_audi_pre$`Puntaje item 2_pre` =
  mem_audi_pre$`Aciertos item 2_pre` - mem_audi_pre$`Errores item 2_pre`

mem_audi_pre$`Puntaje item 1_pre` =
  apply(mem_audi_pre[, "Puntaje item 1_pre"], 1,
        function(x) change_to_zero(x))

mem_audi_pre$`Puntaje item 2_pre` =
  apply(mem_audi_pre[, "Puntaje item 2_pre"], 1,
        function(x) change_to_zero(x))

# Post

```

```

mem_audi_post = read_sheet(url_post, sheet = "Memoria Audi")
vector = c(colnames(mem_audi_post)[1:6], as.vector(unlist(mem_audi_post[1,][7:30])))
colnames(mem_audi_post) = vector
mem_audi_post = mem_audi_post[-1,1:length(vector)]
mem_audi_post = filter(mem_audi_post,
                        !is.na(`Código`),
                        !is.na(Caballo),
                        Caballo != 'NULL')

item_1_post = c("Caballo", "Gallina", "Conejo",
                "Leopardo", "Serpiente", "Cerdo", "Cangrejo", "Leon")

item_2_post = c("Ballena", "Gato", "Abeja",
                "Cebra", "Rana", "Cocodrilo", "Tortuga", "Iguana")

incorrectos_1_post = c("Cabra", "Avestruz", "Hormiga", "Perro")
incorrectos_2_post = c("Toro", "Pulpo", "Loro", "Camello")

mem_audi_post[,7:30] = apply(mem_audi_post[,7:30], 2,
                             function(x) str_replace_all(x, c('NULL' = '0',
                                                                '0' = '0'))))

mem_audi_post$`Aciertos item 1_post` =
  apply(apply(mem_audi_post[,item_1_post], 2,
              function(x) as.numeric(x)),
        1, function(x) sum(x, na.rm= FALSE))

mem_audi_post$`Errores item 1_post` =
  apply(apply(mem_audi_post[,incorrectos_1_post],
              2, function(x) as.numeric(x)),
        1, function(x) sum(x, na.rm= FALSE))

mem_audi_post$`Aciertos item 2_post` =
  apply(apply(mem_audi_post[,item_2_post], 2,
              function(x) as.numeric(x)),
        1, function(x) sum(x, na.rm= FALSE))

mem_audi_post$`Errores item 2_post` =
  apply(apply(mem_audi_post[,incorrectos_2_post],
              2, function(x) as.numeric(x)),
        1, function(x) sum(x, na.rm= FALSE))

mem_audi_post$`Puntaje item 1_post` =
  mem_audi_post$`Aciertos item 1_post` - mem_audi_post$`Errores item 1_post`

mem_audi_post$`Puntaje item 2_post` =

```

```

mem_audi_post$`Aciertos item 2_post` - mem_audi_post$`Errores item 2_post`

mem_audi_post$`Puntaje item 1_post` =
  apply(mem_audi_post[, "Puntaje item 1_post"],
        1, function(x) change_to_zero(x))

mem_audi_post$`Puntaje item 2_post` =
  apply(mem_audi_post[, "Puntaje item 2_post"],
        1, function(x) change_to_zero(x))

```

3.3.1.1 Alpha Estas preguntas tienen una división importante ya que, cada ítem es en realidad un conjunto de varios estímulos a los que el aspirante contesta, es por esto que, el análisis de ítems se hará de forma separada, mientras que para la puntuación final, si se consideran ambos.

3.3.1.1.1 Pre Item 1

```
data_item_1_pre = apply(mem_audi_pre[, item_1_pre], 2, function(x) as.numeric(x))
```

```
alpha(data_item_1_pre)
```

```
## [1] 0.68393
```

```

for(i in seq(length(colnames(data_item_1_pre)))){
  x = alpha(data_item_1_pre[, -i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(data_item_1_pre)[i]))
}

```

```

## [1] "El índice de confiabilidad cambia a 0.647293298485451 al eliminar el ítem Caballo"
## [1] "El índice de confiabilidad cambia a 0.668586459550906 al eliminar el ítem Perro"
## [1] "El índice de confiabilidad cambia a 0.646259126594349 al eliminar el ítem Aguila"
## [1] "El índice de confiabilidad cambia a 0.644089536023538 al eliminar el ítem Pollito"
## [1] "El índice de confiabilidad cambia a 0.65039214278942 al eliminar el ítem Foca"
## [1] "El índice de confiabilidad cambia a 0.663432520864259 al eliminar el ítem Vaca"
## [1] "El índice de confiabilidad cambia a 0.651457547729465 al eliminar el ítem Cocodrilo"
## [1] "El índice de confiabilidad cambia a 0.663303670230139 al eliminar el ítem Sapo"

```

Item 2

```
data_item_2_pre = apply(mem_audi_pre[, item_2_pre], 2, function(x) as.numeric(x))
```

```
alpha(data_item_2_pre)
```

```
## [1] 0.63504
```

```

for(i in seq(length(colnames(data_item_2_pre)))){
  x = alpha(data_item_2_pre[, -i])
  print(paste("El índice de confiabilidad cambia a", x,

```

```

        "al eliminar el ítem",
        colnames(data_item_2_pre)[i]))
}

## [1] "El índice de confiabilidad cambia a 0.602523225795044 al eliminar el ítem Oveja"
## [1] "El índice de confiabilidad cambia a 0.599406712624937 al eliminar el ítem Elefante"
## [1] "El índice de confiabilidad cambia a 0.602042988532722 al eliminar el ítem Tiburon"
## [1] "El índice de confiabilidad cambia a 0.593276625641412 al eliminar el ítem Caracol"
## [1] "El índice de confiabilidad cambia a 0.600396557903299 al eliminar el ítem Ratón"
## [1] "El índice de confiabilidad cambia a 0.629572319619698 al eliminar el ítem Gato"
## [1] "El índice de confiabilidad cambia a 0.596217539674512 al eliminar el ítem Tortuga"
## [1] "El índice de confiabilidad cambia a 0.604201431875818 al eliminar el ítem Pez"

```

3.3.1.1.2 Post Item 1

```
data_item_1_post = apply(mem_audi_post[,item_1_post], 2, function(x) as.numeric(x))
```

```
alpha(data_item_1_post)
```

```

## [1] 0.68693
for(i in seq(length(colnames(data_item_1_post)))){
  x = alpha(data_item_1_post[,-i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(data_item_1_post)[i]))
}

## [1] "El índice de confiabilidad cambia a 0.648096162370693 al eliminar el ítem Caballo"
## [1] "El índice de confiabilidad cambia a 0.655095712431907 al eliminar el ítem Gallina"
## [1] "El índice de confiabilidad cambia a 0.641706952417517 al eliminar el ítem Conejo"
## [1] "El índice de confiabilidad cambia a 0.662406362316556 al eliminar el ítem Leopardo"
## [1] "El índice de confiabilidad cambia a 0.661890660914239 al eliminar el ítem Serpiente"
## [1] "El índice de confiabilidad cambia a 0.656755789628216 al eliminar el ítem Cerdo"
## [1] "El índice de confiabilidad cambia a 0.656994863799219 al eliminar el ítem Cangrejo"
## [1] "El índice de confiabilidad cambia a 0.676666487368243 al eliminar el ítem Leon"

```

Item 2

```
data_item_2_post = apply(mem_audi_post[,item_2_post], 2, function(x) as.numeric(x))
```

```
alpha(data_item_2_post)
```

```

## [1] 0.59202
for(i in seq(length(colnames(data_item_2_post)))){
  x = alpha(data_item_2_post[,-i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(data_item_2_post)[i]))
}

```

```
## [1] "El índice de confiabilidad cambia a 0.542019707351589 al eliminar el ítem Ballena"
## [1] "El índice de confiabilidad cambia a 0.609607253115483 al eliminar el ítem Gato"
## [1] "El índice de confiabilidad cambia a 0.554639606157417 al eliminar el ítem Abeja"
## [1] "El índice de confiabilidad cambia a 0.553631712797516 al eliminar el ítem Cebra"
## [1] "El índice de confiabilidad cambia a 0.534937834410285 al eliminar el ítem Rana"
## [1] "El índice de confiabilidad cambia a 0.543466545267926 al eliminar el ítem Cocodrilo"
## [1] "El índice de confiabilidad cambia a 0.574330362900438 al eliminar el ítem Tortuga"
## [1] "El índice de confiabilidad cambia a 0.558297600405566 al eliminar el ítem Iguana"
```

```
# Amacemos los valores
```

```
alfa = rbind(alfa,
              c("Memoria auditiva (Item 1) - Pre",
                alpha(data_item_1_pre)))

alfa = rbind(alfa,
              c("Memoria auditiva (Item 2) - Pre",
                alpha(data_item_2_pre)))

alfa = rbind(alfa,
              c("Memoria auditiva (Item 1) - Post",
                alpha(data_item_1_post)))

alfa = rbind(alfa,
              c("Memoria auditiva (Item 2) - Post",
                alpha(data_item_2_post)))
```

3.3.1.2 Indicadores psicométricos

3.3.1.2.1 Pre Item 1

```
analitem = item.exam(data_item_1_pre, discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
```

```
style = list(
  cell_text(align="center")
),
locations = cells_body())
```

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.375	0.566	0.415	0.831	0.388	0.212
0.455	0.514	0.314	0.708	0.526	0.234
0.445	0.587	0.407	0.728	0.531	0.261
0.400	0.583	0.424	0.800	0.464	0.233
0.475	0.585	0.390	0.656	0.590	0.278
0.459	0.534	0.336	0.699	0.515	0.245
0.433	0.565	0.386	0.750	0.488	0.245
0.486	0.549	0.341	0.619	0.590	0.267

```
psicometricos_pre = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Memoria auditiva (Item 1) - Pre") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_pre)
```

Item 2

```
analitem = item.exam(data_item_2_pre, discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body())
```

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.455	0.532	0.332	0.707	0.493	0.242
0.496	0.558	0.344	0.568	0.642	0.276
0.435	0.525	0.335	0.747	0.453	0.228
0.467	0.564	0.365	0.679	0.558	0.263
0.449	0.536	0.340	0.721	0.501	0.240
0.499	0.474	0.240	0.527	0.585	0.237
0.442	0.546	0.357	0.734	0.482	0.242
0.428	0.516	0.327	0.759	0.423	0.221

```
psicometricos_pre = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Memoria auditiva (Item 2) - Pre") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_pre)
```

3.3.1.2.2 Post Item 1

```
analitem = item.exam(data_item_1_post, discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body()
```

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.365	0.578	0.433	0.842	0.359	0.211
0.439	0.568	0.387	0.739	0.490	0.249

0.418	0.609	0.448	0.775	0.490	0.254
0.465	0.554	0.357	0.683	0.538	0.258
0.419	0.536	0.357	0.776	0.451	0.225
0.466	0.573	0.380	0.683	0.571	0.267
0.475	0.577	0.380	0.656	0.636	0.274
0.460	0.502	0.297	0.697	0.468	0.231

```
psicometricos_post = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Memoria auditiva (Item 1) - Post") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_post)
```

Item 2

```
analitem = item.exam(data_item_2_post, discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body()
```

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.353	0.535	0.377	0.857	0.333	0.189
0.500	0.402	0.141	0.503	0.448	0.201
0.497	0.542	0.309	0.555	0.650	0.269
0.490	0.541	0.312	0.599	0.585	0.265
0.480	0.581	0.367	0.642	0.585	0.279
0.412	0.539	0.352	0.787	0.431	0.222
0.461	0.472	0.245	0.694	0.476	0.218

0.412	0.495	0.299	0.787	0.384	0.204
-------	-------	-------	-------	-------	-------

```
psicometricos_post = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Memoria auditiva (Item 2) - Post") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_post)
```

```
mem_audi_pre$Total_pre = min_max_scale(mem_audi_pre$`Puntaje item 2_pre` +
                                         mem_audi_pre$`Puntaje item 1_pre`)

mem_audi_post$Total_post = min_max_scale(mem_audi_post$`Puntaje item 2_post` +
                                          mem_audi_post$`Puntaje item 1_post`)

pre_post = inner_join(mem_audi_post,
                      dplyr::select(mem_audi_pre, c("Código", "Total_pre")),
                      by = "Código")
```

```
pivot_to_bind = dplyr::select(pre_post, c("Código",
                                           Total_pre,
                                           Total_post))

pivot_to_bind[,2:ncol(pivot_to_bind)] =
  apply(pivot_to_bind[,2:ncol(pivot_to_bind)],
        2, function(x) 1 - min_max_scale(x))

pivot_to_bind =
  pivot_to_bind %>%
  pivot_longer(cols = !`Código`,
              names_to = c("Prueba", "Tipo"),
              names_sep = "_",
              values_to = "score")

pivot_to_bind$Prueba = str_replace_all(pivot_to_bind$Prueba,
                                       c("Total" = "Memoria auditiva"))

pivot_final = rbind(pivot_final, pivot_to_bind)
```

3.3.1.3 Comparación pre-post

```
print("Estadístico de normalidad pre")
```

3.3.1.3.1 Estadísticos de normalidad

```
## [1] "Estadístico de normalidad pre"
lillie.test(pre_post$Total_pre)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_pre
## D = 0.114, p-value <0.0000000000000002
print("Estadístico de normalidad post")

## [1] "Estadístico de normalidad post"
lillie.test(pre_post$Total_post)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_post
## D = 0.14, p-value <0.0000000000000002

summ = data.frame(unclass(psych::describe(pre_post[,c("Total_pre", "Total_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Total_pre", "Total_post")

summ[,1:10] %>%
  gt()
```

3.3.1.3.2 Descriptivos

vars	n	mean	sd	median	trimmed	mad	min	max	range
Total_pre	886	0.64285	0.22389	0.6875	0.66285	0.18532	0	1	1
Total_post	886	0.67128	0.20887	0.6875	0.69173	0.18532	0	1	1

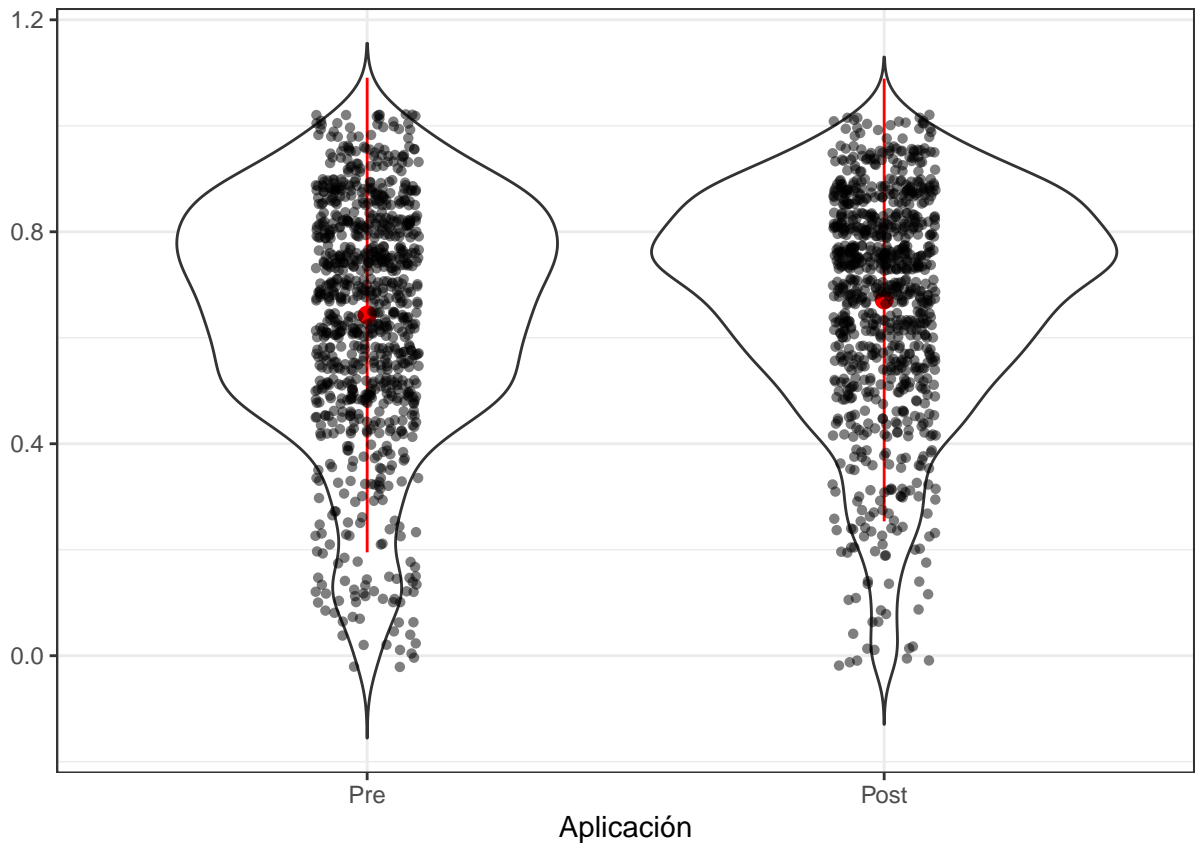
```
comparacion =
  wilcox.test(
    x      = pre_post$Total_pre,
    y      = pre_post$Total_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )
```

comparacion

3.3.1.3.3 Comparación de medias

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Total_pre and pre_post$Total_post
## V = 127996, p-value = 0.00025
## alternative hypothesis: true location shift is not equal to 0

pre_post %>%
  dplyr::select(Total_pre, Total_post) %>%
  pivot_longer(cols = c(Total_pre, Total_post),
               values_to = "value",
               names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
  scale_x_discrete(labels = c("Pre", "Post"))
```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/memoria_auditiva.png")
```

```
size_effect =
  cohen.d(pre_post$Total_post,
          pre_post$Total_pre, paired = TRUE)

size_effect
```

3.3.1.3.4 Tamaño del efecto

```
##
## Cohen's d
##
## d estimate: 0.13119 (negligible)
## 95 percent confidence interval:
##   lower   upper
## 0.060058 0.202314
```

```
# Separamos los descriptivos importantes
```

```
desc = summ[,1:10] %>%
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
```

```

mutate(Prueba = "Memoria auditiva") %>%
mutate(Area = "Funciones ejecutivas") %>%
mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Memoria auditiva",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)

```

3.3.2 Memoria visual

```

# Pre

mem_vis_pre = read_sheet(url_pre, sheet = "Memoria Vis")
vector = c(colnames(mem_vis_pre)[1:6], as.vector(unlist(mem_vis_pre[1,][7:18])))
colnames(mem_vis_pre) = vector
mem_vis_pre = mem_vis_pre[-1,1:length(vector)]
mem_vis_pre = filter(mem_vis_pre,
  !is.na(`Código`),
  !is.na(GYM),
  GYM != 'NULL')

mem_vis_pre[,7:18] = apply(mem_vis_pre[,7:18], 2,
  function(x) str_replace_all(x,'NULL', '0'))

item_1_pre = c("OFICINA","LAVANDERIA","CINE")
item_2_pre = c("BARCO","JUGUETES","ELECTRONICOS")

incorrectos_1_pre = c("GYM", "TALLER", "TIENDA")
incorrectos_2_pre = c("SALON BELLEZA", "AEROPUERTO", "IGLESIA")

mem_vis_pre$`Aciertos item 1_pre` =
  apply(apply(mem_vis_pre[,item_1_pre], 2,
    function(x) as.numeric(x)),
    1, function(x) sum(x, na.rm= FALSE))

mem_vis_pre$`Errores item 1_pre` =
  apply(apply(mem_vis_pre[,incorrectos_1_pre], 2,
    function(x) as.numeric(x)),

```

```

1, function(x) sum(x, na.rm= FALSE))

mem_vis_pre$`Aciertos item 2_pre` =
  apply(apply(mem_vis_pre[,item_2_pre], 2,
             function(x) as.numeric(x)),
        1, function(x) sum(x, na.rm= FALSE))

mem_vis_pre$`Errores item 2_pre` =
  apply(apply(mem_vis_pre[,incorrectos_2_pre], 2,
             function(x) as.numeric(x)),
        1, function(x) sum(x, na.rm= FALSE))

mem_vis_pre$`Puntaje item 1_pre` =
  mem_vis_pre$`Aciertos item 1_pre` - mem_vis_pre$`Errores item 1_pre`

mem_vis_pre$`Puntaje item 2_pre` =
  mem_vis_pre$`Aciertos item 2_pre` - mem_vis_pre$`Errores item 2_pre`

mem_vis_pre$`Puntaje item 1_pre` =
  apply(mem_vis_pre[, "Puntaje item 1_pre"], 1,
        function(x) change_to_zero(x))

mem_vis_pre$`Puntaje item 2_pre` =
  apply(mem_vis_pre[, "Puntaje item 2_pre"], 1,
        function(x) change_to_zero(x))

## Post

mem_vis_post = read_sheet(url_post, sheet = "Memoria Vis")
vector = c(colnames(mem_vis_post)[1:6], as.vector(unlist(mem_vis_post[1,][7:18])))
colnames(mem_vis_post) = vector

mem_vis_post = mem_vis_post[-1,1:length(vector)]
mem_vis_post = filter(mem_vis_post,
                      !is.na(`Código`),
                      # !is.null(Cancha),
                      Campo != 'NULL')

mem_vis_post[,7:18] = apply(mem_vis_post[,7:18], 2,
                           function(x) str_replace_all(x,'NULL', '0'))

item_1_post = c("Campo",      "Colegio_", "Noira")
item_2_post = c("Heladeria",  "Parque",  "Aeropuerto")

incorrectos_1_post = c("Iglesia", "GYM", "Zoo")
incorrectos_2_post = c("Bus stop", "Playa", "Frutas")

```

```

mem_vis_post$`Aciertos item 1_post` =
  apply(apply(mem_vis_post[,item_1_post], 2,
              function(x) as.numeric(x)),
        1, function(x) sum(x, na.rm= FALSE))

mem_vis_post$`Errores item 1_post` =
  apply(apply(mem_vis_post[,incorrectos_1_post],
              2, function(x) as.numeric(x)),
        1, function(x) sum(x, na.rm= FALSE))

mem_vis_post$`Aciertos item 2_post` =
  apply(apply(mem_vis_post[,item_2_post], 2,
              function(x) as.numeric(x)),
        1, function(x) sum(x, na.rm= FALSE))

mem_vis_post$`Errores item 2_post` =
  apply(apply(mem_vis_post[,incorrectos_2_post],
              2, function(x) as.numeric(x)),
        1, function(x) sum(x, na.rm= FALSE))

mem_vis_post$`Puntaje item 1_post` =
  mem_vis_post$`Aciertos item 1_post` - mem_vis_post$`Errores item 1_post`

mem_vis_post$`Puntaje item 2_post` =
  mem_vis_post$`Aciertos item 2_post` - mem_vis_post$`Errores item 2_post`

mem_vis_post$`Puntaje item 1_post` =
  apply(mem_vis_post[, "Puntaje item 1_post"], 1,
        function(x) change_to_zero(x))

mem_vis_post$`Puntaje item 2_post` =
  apply(mem_vis_post[, "Puntaje item 2_post"], 1,
        function(x) change_to_zero(x))

```

3.3.2.1 Alpha Estas preguntas tienen una división importante ya que, cada ítem es en realidad un conjunto de varios estímulos a los que el aspirante contesta, es por esto que, el análisis de ítems se hará de forma separada, mientras que para la puntuación final, si se consideran ambos.

3.3.2.1.1 Pre Item 1

```

data_item_1_pre = apply(mem_vis_pre[,item_1_pre], 2,
                        function(x) as.numeric(x))

```

```

alpha(data_item_1_pre)

```

```

## [1] 0.47356

```



```
for(i in seq(length(colnames(data_item_1_pre)))){
  x = alpha(data_item_1_pre[,-i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(data_item_1_pre)[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.388447002273224 al eliminar el ítem OFICINA"
## [1] "El índice de confiabilidad cambia a 0.315482725073857 al eliminar el ítem LAVANDERIA"
## [1] "El índice de confiabilidad cambia a 0.412568830952269 al eliminar el ítem CINE"
```

Item 2

```
data_item_2_pre = apply(mem_vis_pre[,item_2_pre], 2, function(x) as.numeric(x))

alpha(data_item_2_pre)
```

```
## [1] 0.43212
```

```
for(i in seq(length(colnames(data_item_2_pre)))){
  x = alpha(data_item_2_pre[,-i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(data_item_2_pre)[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.37985401459854 al eliminar el ítem BARCO"
## [1] "El índice de confiabilidad cambia a 0.287390972868244 al eliminar el ítem JUGUETES"
## [1] "El índice de confiabilidad cambia a 0.335037117006946 al eliminar el ítem ELECTRONICOS"
```

3.3.2.1.2 Post Item 1

```
data_item_1_post = apply(mem_vis_post[,item_1_post], 2,
                          function(x) as.numeric(x))

alpha(data_item_1_post)
```

```
## [1] 0.43532
```

```
for(i in seq(length(colnames(data_item_1_post)))){
  x = alpha(data_item_1_post[,-i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(data_item_1_post)[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.274206722509196 al eliminar el ítem Campo"
## [1] "El índice de confiabilidad cambia a 0.435664211177702 al eliminar el ítem Colegio_"
## [1] "El índice de confiabilidad cambia a 0.341113801452785 al eliminar el ítem Noira"
```

Item 2

```

data_item_2_post = apply(mem_vis_post[,item_2_post], 2,
                        function(x) as.numeric(x))

alpha(data_item_2_post)

## [1] 0.35062

for(i in seq(length(colnames(data_item_2_post)))){
  x = alpha(data_item_2_post[,i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(data_item_2_post)[i]))
}

## [1] "El índice de confiabilidad cambia a 0.279144916728809 al eliminar el ítem Heladeria"
## [1] "El índice de confiabilidad cambia a 0.179232291935623 al eliminar el ítem Parque"
## [1] "El índice de confiabilidad cambia a 0.324197189869957 al eliminar el ítem Aeropuerto"

# Amacenas los valores

alfa = rbind(alfa,
             c("Memoria visual (Item 1) - Pre",
               alpha(data_item_1_pre)))

alfa = rbind(alfa,
             c("Memoria visual (Item 2) - Pre",
               alpha(data_item_2_pre)))

alfa = rbind(alfa,
             c("Memoria visual (Item 1) - Post",
               alpha(data_item_1_post)))

alfa = rbind(alfa,
             c("Memoria visual (Item 2) - Post",
               alpha(data_item_2_post)))

```

3.3.2.2 Indicadores psicométricos

3.3.2.2.1 Pre Item 1

```

analitem = item.exam(data_item_1_pre, discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  )

```

```

) %>%
tab_style(
  style = list(
    cell_text(align="center", weight="bold")
  ),
  locations=cells_column_labels()
)%>%
tab_style(
  style = list(
    cell_text(align="center")
  ),
  locations = cells_body())

```

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.434	0.732	0.291	0.748	0.620	0.318
0.405	0.727	0.327	0.794	0.561	0.294
0.339	0.632	0.274	0.867	0.367	0.214

```

psicometricos_pre = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Memoria visual (Item 1) - Pre") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_pre)

```

Item 2

```

analitem = item.exam(data_item_2_pre, discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  )%>%
  tab_style(
    style = list(
      cell_text(align="center")
    )
  )

```

```
),
locations = cells_body())
```

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.434	0.648	0.236	0.748	0.566	0.281
0.482	0.721	0.286	0.634	0.779	0.347
0.453	0.682	0.262	0.711	0.647	0.309

```
psicometricos_pre = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Memoria visual (Item 2) - Pre") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_pre)
```

3.3.2.2.2 Post Item 1

```
analitem = item.exam(data_item_1_post, discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body())
```

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.227	0.627	0.348	0.948	0.162	0.142
0.397	0.753	0.237	0.804	0.588	0.299
0.331	0.691	0.263	0.875	0.370	0.229

```
psicometricos_post = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Memoria visual (Item 1) - Post") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_post)
```

Item 2

```
analitem = item.exam(data_item_2_post, discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body())
```

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.366	0.621	0.194	0.841	0.398	0.227
0.420	0.710	0.240	0.771	0.557	0.298
0.402	0.645	0.172	0.797	0.515	0.259

```
psicometricos_post = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Memoria visual (Item 2) - Post") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_post)
```

```

mem_vis_pre$Total_pre = min_max_scale(mem_vis_pre$`Puntaje item 2_pre` +
                                     mem_vis_pre$`Puntaje item 1_pre`)

mem_vis_post$Total_post = min_max_scale(mem_vis_post$`Puntaje item 2_post` +
                                     mem_vis_post$`Puntaje item 1_post`)

pre_post = inner_join(mem_vis_post,
                      dplyr::select(mem_vis_pre, c("Código", "Total_pre")),
                      by = "Código")

```

```

pivot_to_bind = dplyr::select(pre_post, c("Código",
                                           Total_pre,
                                           Total_post))

pivot_to_bind[,2:ncol(pivot_to_bind)] =
  apply(pivot_to_bind[,2:ncol(pivot_to_bind)],
        2, function(x) 1 - min_max_scale(x))

pivot_to_bind =
  pivot_to_bind %>%
  pivot_longer(cols = !`Código`,
              names_to = c("Prueba", "Tipo"),
              names_sep = "_",
              values_to = "score")

pivot_to_bind$Prueba = str_replace_all(pivot_to_bind$Prueba,
                                       c("Total" = "Memoria visual"))

pivot_final = rbind(pivot_final, pivot_to_bind)

```

3.3.2.3 Comparación pre-post

```
print("Estadístico de normalidad pre")
```

3.3.2.3.1 Estadísticos de normalidad

```
## [1] "Estadístico de normalidad pre"
```

```
lillie.test(pre_post$Total_pre)
```

```

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_pre
## D = 0.139, p-value <0.0000000000000002

```

```
print("Estadístico de normalidad post")

## [1] "Estadístico de normalidad post"
lillie.test(pre_post$Total_post)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_post
## D = 0.193, p-value <0.0000000000000002
```

```
summ = data.frame(unclass(psych::describe(pre_post[,c("Total_pre",
                                                    "Total_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Total_pre", "Total_post")

summ[,1:10] %>%
  gt()
```

3.3.2.3.2 Descriptivos

vars	n	mean	sd	median	trimmed	mad	min	max	range
Total_pre	887	0.58437	0.32135	0.66667	0.60525	0.2471	0	1	1
Total_post	887	0.68828	0.30195	0.83333	0.72621	0.2471	0	1	1

```
comparacion =
  wilcox.test(
    x = pre_post$Total_pre,
    y = pre_post$Total_post,
    alternative = "two.sided",
    mu = 0,
    var.equal = TRUE,
    paired = TRUE,
    conf.level = 0.95
  )

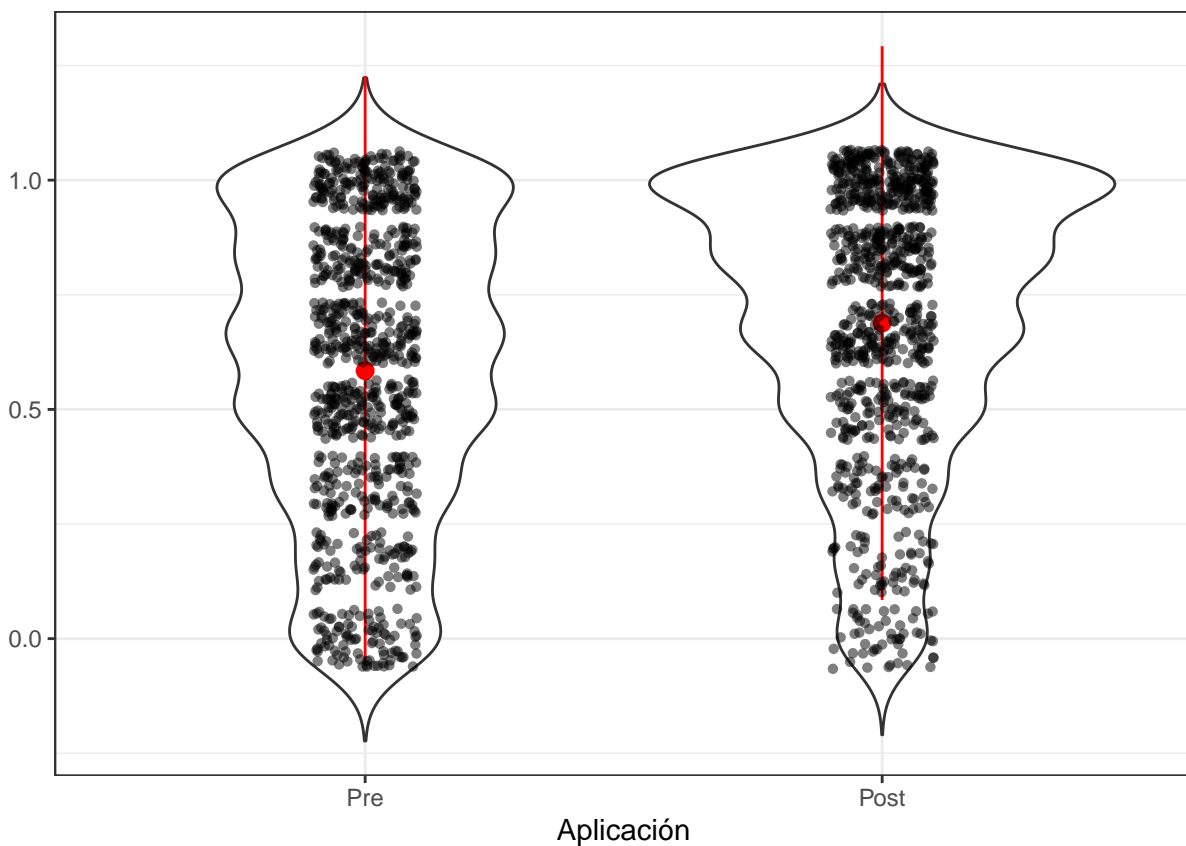
comparacion
```

3.3.2.3.3 Comparación de medias

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  pre_post$Total_pre and pre_post$Total_post
```

```
## V = 69519, p-value <0.0000000000000002
## alternative hypothesis: true location shift is not equal to 0
```

```
pre_post %>%
  dplyr::select(Total_pre, Total_post) %>%
  pivot_longer(cols = c(Total_pre, Total_post),
               values_to = "value",
               names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
  scale_x_discrete(labels = c("Pre", "Post"))
```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/memoria_visual.png")
```

```
size_effect =
  cohen.d(pre_post$Total_post,
```



```
pre_post$Total_pre, paired = TRUE)

size_effect
```

3.3.2.3.4 Tamaño del efecto

```
##
## Cohen's d
##
## d estimate: 0.33305 (small)
## 95 percent confidence interval:
## lower upper
## 0.25791 0.40820

# Separamos los descriptivos importantes
desc = summ[,1:10] %>%
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
  mutate(Prueba = "Memoria visual") %>%
  mutate(Area = "Funciones ejecutivas") %>%
  mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Memoria visual",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)
```

3.3.3 Inhibición

```
# Pre

inhibicion_pre = read_sheet(url_pre, sheet = "Inhibición")

col_items = paste(rep("Item", 20), rep(1:2, each = 10),
  rep(paste("_", rep(1:10, 2))))

vector = c(colnames(inhibicion_pre)[1:6], col_items)
colnames(inhibicion_pre) = vector

inhibicion_pre = inhibicion_pre[,1:26]
```

```

inhibicion_pre = filter(inhibicion_pre,
                        !is.na(`Código`),
                        !is.na(`Item 1 _ 1`))

inhibicion_pre[,col_items] = apply(inhibicion_pre[,col_items], 2,
                                   function(x) str_to_upper(x))

claves_inhibicion_pre = c('N','M','M','N','M','N','M',
                          'N','M','N','N','M','N','M','N',
                          'M','M','N','M','N')

inhibicion_pre[,7:26] = calificacion(inhibicion_pre[,7:26], claves_inhibicion_pre)

inhibicion_pre$`Aciertos item 1_pre` = apply(inhibicion_pre[,7:16], 1,
                                             function(x) sum(x, na.rm= FALSE))

inhibicion_pre$`Aciertos item 2_pre` = apply(inhibicion_pre[,17:26], 1,
                                             function(x) sum(x, na.rm= FALSE))

# Post

inhibicion_post = read_sheet(url_post, sheet = "Inhibición")

col_items = paste(rep("Item", 20), rep(1:2, each = 10),
                  rep(paste("_", rep(1:10, 2))))

vector = c(colnames(inhibicion_post)[1:6], col_items)
colnames(inhibicion_post) = vector

inhibicion_post = inhibicion_post[,1:26]

inhibicion_post = filter(inhibicion_post,
                        !is.na(`Código`),
                        !is.na(`Item 1 _ 1`))

inhibicion_post[,col_items] = apply(inhibicion_post[,col_items], 2,
                                   function(x) str_to_upper(x))

claves_inhibicion_post = c('E','S','S','E','S','E','E','S','E','S',
                          'S','S','E','S','E','S','E','S','E','E')

inhibicion_post[,7:26] = calificacion(inhibicion_post[,7:26], claves_inhibicion_post)

inhibicion_post$`Aciertos item 1_post` = apply(inhibicion_post[,7:16], 1,

```

```

                                function(x) sum(x, na.rm= FALSE))
inhibicion_post$`Aciertos item 2_post` = apply(inhibicion_post[,17:26], 1,
                                function(x) sum(x, na.rm= FALSE))

```

3.3.3.1 Alpha

```

data_item_2_pre = apply(inhibicion_pre[,17:26], 2,
                        function(x) as.numeric(x))

```

```

alpha(data_item_2_pre)

```

3.3.3.1.1 Pre

```
## [1] 0.93246
```

```

for(i in seq(length(colnames(data_item_2_pre)))){
  x = alpha(data_item_2_pre[,-i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(data_item_2_pre)[i]))
}

```

```

## [1] "El índice de confiabilidad cambia a 0.928742182564455 al eliminar el ítem Item 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.926339546605565 al eliminar el ítem Item 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.922125410565513 al eliminar el ítem Item 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.92322129697545 al eliminar el ítem Item 2 _ 4"
## [1] "El índice de confiabilidad cambia a 0.923712817380478 al eliminar el ítem Item 2 _ 5"
## [1] "El índice de confiabilidad cambia a 0.924159564622178 al eliminar el ítem Item 2 _ 6"
## [1] "El índice de confiabilidad cambia a 0.929238349202597 al eliminar el ítem Item 2 _ 7"
## [1] "El índice de confiabilidad cambia a 0.92423950158183 al eliminar el ítem Item 2 _ 8"
## [1] "El índice de confiabilidad cambia a 0.926142282467351 al eliminar el ítem Item 2 _ 9"
## [1] "El índice de confiabilidad cambia a 0.926896558740839 al eliminar el ítem Item 2 _ 10"

```

```

data_item_2_post = apply(inhibicion_post[,17:26], 2,
                        function(x) as.numeric(x))

```

```

alpha(data_item_2_post)

```

3.3.3.1.2 Post

```
## [1] 0.94096
```

```

for(i in seq(length(colnames(data_item_2_post)))){
  x = alpha(data_item_2_post[,-i])
  print(paste("El índice de confiabilidad cambia a", x,

```

```

        "al eliminar el ítem",
        colnames(data_item_2_post)[i]))
}

## [1] "El índice de confiabilidad cambia a 0.938602087099249 al eliminar el ítem Item 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.935904184135684 al eliminar el ítem Item 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.935105936272074 al eliminar el ítem Item 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.934894883841978 al eliminar el ítem Item 2 _ 4"
## [1] "El índice de confiabilidad cambia a 0.930769011003884 al eliminar el ítem Item 2 _ 5"
## [1] "El índice de confiabilidad cambia a 0.932925017615792 al eliminar el ítem Item 2 _ 6"
## [1] "El índice de confiabilidad cambia a 0.932903499969362 al eliminar el ítem Item 2 _ 7"
## [1] "El índice de confiabilidad cambia a 0.934858607806564 al eliminar el ítem Item 2 _ 8"
## [1] "El índice de confiabilidad cambia a 0.933555372890058 al eliminar el ítem Item 2 _ 9"
## [1] "El índice de confiabilidad cambia a 0.938406866873823 al eliminar el ítem Item 2 _ 10"

# Amacemos los valores

alfa = rbind(alfa,
              c("Inhibición - Pre",
                alpha(data_item_2_pre)))

alfa = rbind(alfa,
              c("Inhibición - Post",
                alpha(data_item_2_post)))

```

3.3.3.2 Indicadores psicométricos

```

analitem = item.exam(data_item_2_pre, discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
  )

```

```
locations = cells_body()
```

3.3.3.2.1 Pre

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.364	0.730	0.666	0.843	0.450	0.266
0.374	0.773	0.716	0.832	0.493	0.289
0.388	0.842	0.798	0.816	0.544	0.326
0.391	0.825	0.777	0.812	0.542	0.323
0.395	0.818	0.767	0.806	0.558	0.323
0.378	0.810	0.760	0.827	0.507	0.306
0.397	0.731	0.661	0.804	0.553	0.290
0.384	0.809	0.758	0.821	0.520	0.310
0.397	0.781	0.721	0.804	0.569	0.310
0.384	0.766	0.705	0.821	0.520	0.294

```
psicometricos_post = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Inhibición - Pre") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_post)
```

```
analitem = item.exam(data_item_2_post, discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body()
```

3.3.3.2.2 Post

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.273	0.730	0.675	0.919	0.244	0.199
0.311	0.787	0.735	0.892	0.325	0.245
0.339	0.807	0.753	0.868	0.398	0.273
0.332	0.809	0.757	0.874	0.378	0.268
0.323	0.877	0.843	0.882	0.356	0.283
0.324	0.841	0.798	0.881	0.359	0.273
0.335	0.842	0.798	0.871	0.387	0.282
0.330	0.809	0.757	0.876	0.373	0.267
0.320	0.830	0.785	0.884	0.347	0.265
0.342	0.753	0.687	0.865	0.406	0.258

```
psicometricos_post = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Inhibición - Post") %>%
  mutate(Item = row.names(.))
```

```
indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_post)
```

```
inhibicion_pre$Total_pre = min_max_scale(inhibicion_pre$`Aciertos item 2_pre`)
inhibicion_post$Total_post = min_max_scale(inhibicion_post$`Aciertos item 2_post`)
```

```
pre_post = inner_join(inhibicion_post,
                      dplyr::select(inhibicion_pre, c("Código",
                                                         "Total_pre")),
                      by = "Código")
```

```
pivot_to_bind = dplyr::select(pre_post, c("Código",
                                           Total_pre,
                                           Total_post))
```

```
pivot_to_bind[,2:ncol(pivot_to_bind)] =
  apply(pivot_to_bind[,2:ncol(pivot_to_bind)],
        2, function(x) 1 - min_max_scale(x))
```

```
pivot_to_bind =
  pivot_to_bind %>%
  pivot_longer(cols = !`Código`,
              names_to = c("Prueba", "Tipo"),
              names_sep = "_",
              values_to = "score")
```

```

pivot_to_bind$Prueba = str_replace_all(pivot_to_bind$Prueba,
                                         c("Total" = "Inhibición"))

pivot_final = rbind(pivot_final, pivot_to_bind)

```

3.3.3.3 Comparación pre-post

```
print("Estadístico de normalidad pre")
```

3.3.3.3.1 Estadísticos de normalidad

```
## [1] "Estadístico de normalidad pre"
```

```
lillie.test(pre_post$Total_pre)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_pre
## D = 0.345, p-value <0.0000000000000002
```

```
print("Estadístico de normalidad post")
```

```
## [1] "Estadístico de normalidad post"
```

```
lillie.test(pre_post$Total_post)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_post
## D = 0.41, p-value <0.0000000000000002
```

```

summ = data.frame(unclass(psych::describe(pre_post[,c("Total_pre",
                                                       "Total_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Total_pre", "Total_post")

summ[,1:10] %>%
  gt()

```

3.3.3.3.2 Descriptivos

vars	n	mean	sd	median	trimmed	mad	min	max	range
Total_pre	888	0.82703	0.30236	1	0.89916	0	0	1	1
Total_post	888	0.89088	0.24787	1	0.96152	0	0	1	1

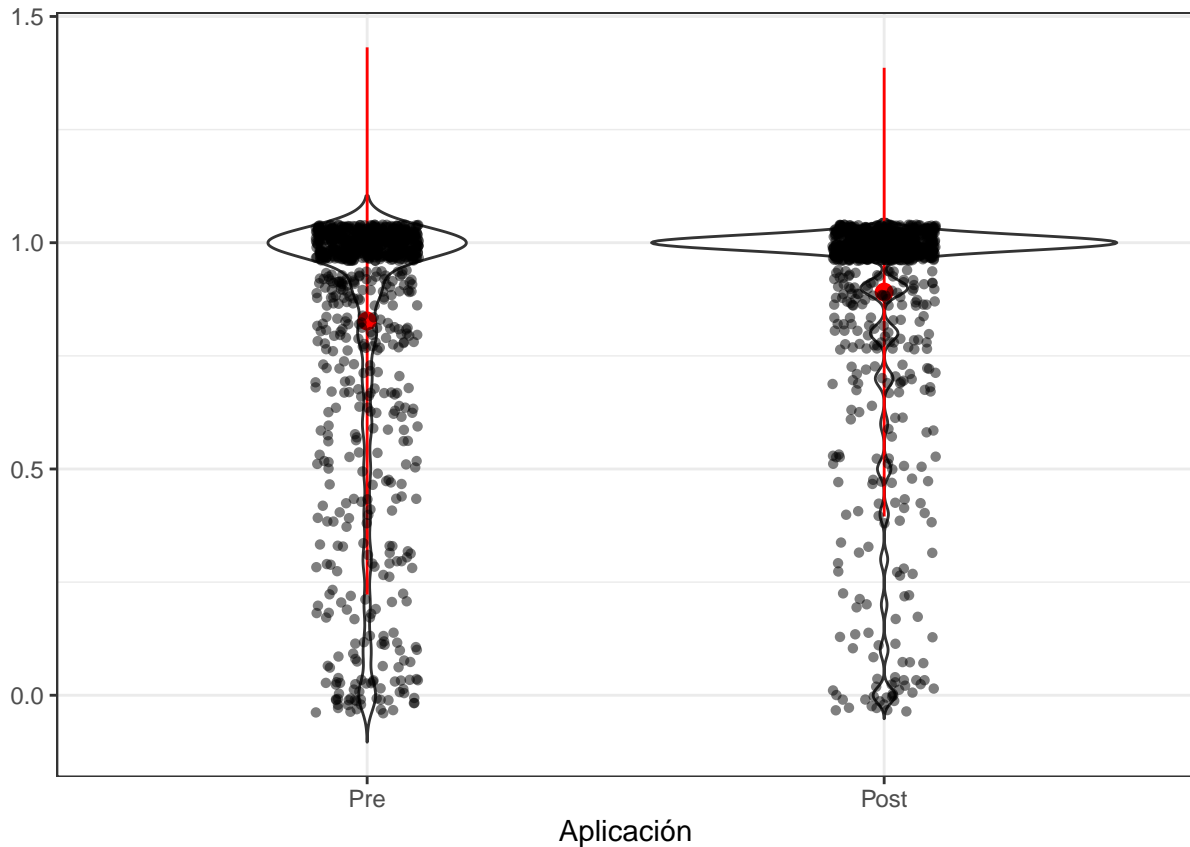
```
comparacion =
  wilcox.test(
    x      = pre_post$Total_pre,
    y      = pre_post$Total_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )
```

```
comparacion
```

3.3.3.3.3 Comparación de medias

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Total_pre and pre_post$Total_post
## V = 26254, p-value = 0.00000000017
## alternative hypothesis: true location shift is not equal to 0
```

```
pre_post %>%
  dplyr::select(Total_pre, Total_post) %>%
  pivot_longer(cols = c(Total_pre, Total_post),
               values_to = "value",
               names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
  scale_x_discrete(labels = c("Pre", "Post"))
```

```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/inhibicion.png")
```

```
size_effect =
  cohen.d(pre_post$Total_post,
          pre_post$Total_pre, paired = TRUE)

size_effect
```

3.3.3.3.4 Tamaño del efecto

```
##
## Cohen's d
##
## d estimate: 0.22972 (small)
## 95 percent confidence interval:
##   lower   upper
## 0.15415 0.30530
```

```
# Separamos los descriptivos importantes
```

```
desc = summ[,1:10] %>%
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
```

```

mutate(Prueba = "Inhibición") %>%
mutate(Area = "Funciones ejecutivas") %>%
mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Inhibición",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)

```

3.3.4 Flexibilidad

```

# Pre

flexibilidad_pre = read_sheet(url_pre, sheet = "Flexibilidad")

col_items = paste(rep("Item", 20), rep(1:2, each = 10),
  rep(paste("_", rep(1:10, 2))))

vector = c(colnames(flexibilidad_pre)[1:6], col_items)

colnames(flexibilidad_pre) = vector

flexibilidad_pre = flexibilidad_pre[,1:26]

flexibilidad_pre = filter(flexibilidad_pre,
  !is.na(`Código`),
  !is.na(`Item 1 _ 1`),
  `Item 1 _ 1` != 'NULL')

flexibilidad_pre[,col_items] = apply(flexibilidad_pre[,col_items], 2,
  function(x) str_to_upper(x))

claves_flexibilidad_pre = c('A','M','N','A','N','M','M','N',
  'A','A','N','3','M','M','3','N',
  'N','M','3','3')

flexibilidad_pre[,7:26] = calificacion(flexibilidad_pre[,7:26],
  claves_flexibilidad_pre)

```

```

flexibilidad_pre$`Aciertos item 1_pre` =
  apply(flexibilidad_pre[,7:16], 1,
        function(x) sum(x, na.rm= FALSE))

flexibilidad_pre$`Aciertos item 2_pre` =
  apply(flexibilidad_pre[,17:26], 1,
        function(x) sum(x, na.rm= FALSE))

# Post

flexibilidad_post = read_sheet(url_post, sheet = "Flexibilidad")

col_items = paste(rep("Item", 20), rep(1:2, each = 10),
                  rep(paste("_", rep(1:10, 2))))

vector = c(colnames(flexibilidad_post)[1:6], col_items)

colnames(flexibilidad_post) = vector

flexibilidad_post = flexibilidad_post[,1:26]

flexibilidad_post = filter(flexibilidad_post,
                           !is.na(`Código`),
                           !is.na(`Item 1 _ 1`),
                           `Item 1 _ 1` != 'NULL')

flexibilidad_post[,col_items] = apply(flexibilidad_post[,col_items], 2,
                                       function(x) str_to_upper(x))

claves_flexibilidad_post = c('H','S','E','E','H','S','H','E','H','S','S',
                             '3','E','E','3','S','S','3','E','S')

flexibilidad_post[,7:26] = calificacion(flexibilidad_post[,7:26],
                                       claves_flexibilidad_post)

flexibilidad_post$`Aciertos item 1_post` =
  apply(flexibilidad_post[,7:16], 1,
        function(x) sum(x, na.rm= FALSE))

flexibilidad_post$`Aciertos item 2_post` =
  apply(flexibilidad_post[,17:26], 1,
        function(x) sum(x, na.rm= FALSE))

```

3.3.4.1 Alpha

```
alpha(flexibilidad_pre[,17:26])
```

3.3.4.1.1 Pre

```
## [1] 0.90982
```

```
for(i in seq(length(colnames(flexibilidad_pre[,17:26])))){  
  x = alpha(flexibilidad_pre[,17:26][,-i])  
  print(paste("El índice de confiabilidad cambia a", x,  
              "al eliminar el ítem",  
              colnames(flexibilidad_pre[,17:26])[i]))  
}
```

```
## [1] "El índice de confiabilidad cambia a 0.904909973180862 al eliminar el ítem Item 2 _ 1"  
## [1] "El índice de confiabilidad cambia a 0.909522941539101 al eliminar el ítem Item 2 _ 2"  
## [1] "El índice de confiabilidad cambia a 0.898025951110559 al eliminar el ítem Item 2 _ 3"  
## [1] "El índice de confiabilidad cambia a 0.895290775574781 al eliminar el ítem Item 2 _ 4"  
## [1] "El índice de confiabilidad cambia a 0.902461975959701 al eliminar el ítem Item 2 _ 5"  
## [1] "El índice de confiabilidad cambia a 0.896121454224979 al eliminar el ítem Item 2 _ 6"  
## [1] "El índice de confiabilidad cambia a 0.89590551320103 al eliminar el ítem Item 2 _ 7"  
## [1] "El índice de confiabilidad cambia a 0.897698955313045 al eliminar el ítem Item 2 _ 8"  
## [1] "El índice de confiabilidad cambia a 0.903299006162163 al eliminar el ítem Item 2 _ 9"  
## [1] "El índice de confiabilidad cambia a 0.903039322152149 al eliminar el ítem Item 2 _ 10"
```

```
alpha(flexibilidad_post[,17:26])
```

3.3.4.1.2 Post

```
## [1] 0.92173
```

```
for(i in seq(length(colnames(flexibilidad_post[,17:26])))){  
  x = alpha(flexibilidad_post[,17:26][,-i])  
  print(paste("El índice de confiabilidad cambia a", x,  
              "al eliminar el ítem",  
              colnames(flexibilidad_post[,17:26])[i]))  
}
```

```
## [1] "El índice de confiabilidad cambia a 0.918060838498837 al eliminar el ítem Item 2 _ 1"  
## [1] "El índice de confiabilidad cambia a 0.916623613781108 al eliminar el ítem Item 2 _ 2"  
## [1] "El índice de confiabilidad cambia a 0.913757508294489 al eliminar el ítem Item 2 _ 3"  
## [1] "El índice de confiabilidad cambia a 0.911732835915132 al eliminar el ítem Item 2 _ 4"  
## [1] "El índice de confiabilidad cambia a 0.916086515616851 al eliminar el ítem Item 2 _ 5"  
## [1] "El índice de confiabilidad cambia a 0.913085929144761 al eliminar el ítem Item 2 _ 6"  
## [1] "El índice de confiabilidad cambia a 0.910564207467452 al eliminar el ítem Item 2 _ 7"  
## [1] "El índice de confiabilidad cambia a 0.915358897837899 al eliminar el ítem Item 2 _ 8"  
## [1] "El índice de confiabilidad cambia a 0.909831420303603 al eliminar el ítem Item 2 _ 9"  
## [1] "El índice de confiabilidad cambia a 0.912152661088202 al eliminar el ítem Item 2 _ 10"
```

```
# Amacemos los valores

alfa = rbind(alfa,
             c("Flexibilidad - Pre",
               alpha(flexibilidad_pre[,17:26])))

alfa = rbind(alfa,
             c("Flexibilidad - Post",
               alpha(flexibilidad_post[,17:26])))
```

3.3.4.2 Indicadores psicométricos

```
analitem = item.exam(flexibilidad_pre[,17:26], discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body()
```

3.3.4.2.1 Pre

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.370	0.691	0.607	0.837	0.488	0.255
0.324	0.601	0.515	0.881	0.356	0.195
0.404	0.789	0.719	0.795	0.606	0.319
0.400	0.818	0.758	0.800	0.571	0.327
0.320	0.714	0.647	0.884	0.345	0.228
0.371	0.805	0.747	0.835	0.488	0.299
0.383	0.809	0.749	0.822	0.520	0.310
0.397	0.790	0.723	0.804	0.566	0.313
0.301	0.700	0.635	0.900	0.299	0.211

0.282	0.707	0.648	0.913	0.259	0.199
-------	-------	-------	-------	-------	-------

```
psicometricos_post = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Flexibilidad - Pre") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_post)
```

```
analitem = item.exam(flexibilidad_post[,17:26], discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body()
```

3.3.4.2.2 Post

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.357	0.704	0.627	0.850	0.445	0.252
0.300	0.715	0.653	0.900	0.294	0.214
0.338	0.765	0.704	0.868	0.389	0.258
0.368	0.798	0.739	0.839	0.476	0.294
0.331	0.727	0.660	0.875	0.373	0.240
0.365	0.779	0.716	0.842	0.468	0.284
0.375	0.815	0.759	0.831	0.496	0.305
0.337	0.740	0.674	0.869	0.389	0.249
0.365	0.824	0.772	0.842	0.468	0.300
0.359	0.791	0.732	0.848	0.448	0.284

```
psicometricos_post = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Flexibilidad - Post") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_post)
```

```
flexibilidad_pre$Total_pre = min_max_scale(flexibilidad_pre$`Aciertos item 2_pre`)
flexibilidad_post$Total_post = min_max_scale(flexibilidad_post$`Aciertos item 2_post`)

pre_post = inner_join(flexibilidad_post,
                      dplyr::select(flexibilidad_pre, c("Código",
                                                         "Total_pre")),
                      by = "Código")
```

```
pivot_to_bind = dplyr::select(pre_post, c("Código",
                                           Total_pre,
                                           Total_post))

pivot_to_bind[,2:ncol(pivot_to_bind)] =
  apply(pivot_to_bind[,2:ncol(pivot_to_bind)],
        2, function(x) 1 - min_max_scale(x))

pivot_to_bind =
  pivot_to_bind %>%
  pivot_longer(cols = !`Código`,
              names_to = c("Prueba", "Tipo"),
              names_sep = "_",
              values_to = "score")

pivot_to_bind$Prueba = str_replace_all(pivot_to_bind$Prueba,
                                       c("Total" = "Flexibilidad"))

pivot_final = rbind(pivot_final, pivot_to_bind)
```

3.3.4.3 Comparación pre-post

```
print("Estadístico de normalidad pre")
```

3.3.4.3.1 Estadísticos de normalidad

```
## [1] "Estadístico de normalidad pre"
```

```

lillie.test(pre_post$Total_pre)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_pre
## D = 0.35, p-value <0.0000000000000002

print("Estadístico de normalidad post")

## [1] "Estadístico de normalidad post"

lillie.test(pre_post$Total_post)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_post
## D = 0.36, p-value <0.0000000000000002

```

```

summ = data.frame(unclass(psych::describe(pre_post[,c("Total_pre",
                                                    "Total_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Total_pre", "Total_post")

summ[,1:10] %>%
  gt()

```

3.3.4.3.2 Descriptivos

vars	n	mean	sd	median	trimmed	mad	min	max	range
Total_pre	887	0.85051	0.26251	1	0.91252	0	0	1	1
Total_post	887	0.86809	0.25715	1	0.93572	0	0	1	1

```

comparacion =
  wilcox.test(
    x      = pre_post$Total_pre,
    y      = pre_post$Total_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )

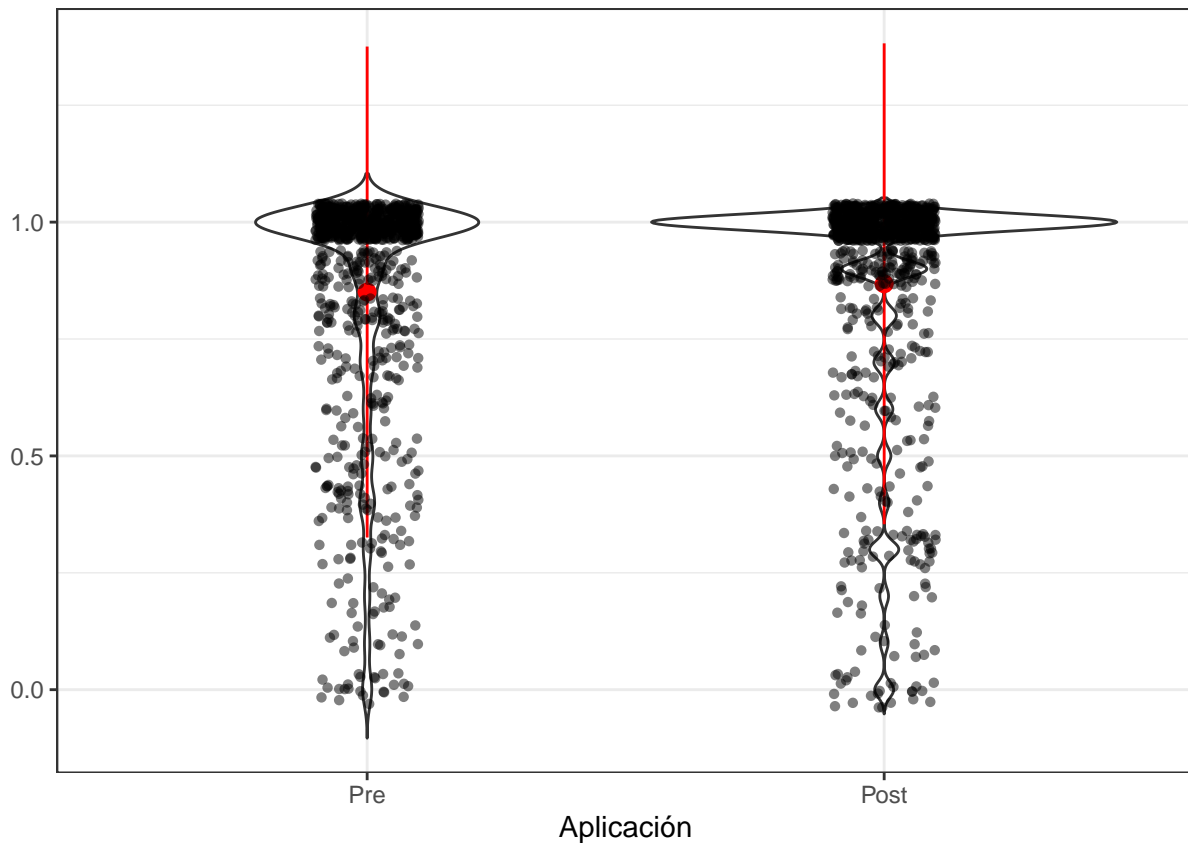
```


comparacion

3.3.4.3.3 Comparación de medias

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Total_pre and pre_post$Total_post
## V = 45124, p-value = 0.049
## alternative hypothesis: true location shift is not equal to 0

pre_post %>%
  dplyr::select(Total_pre, Total_post) %>%
  pivot_longer(cols = c(Total_pre, Total_post),
               values_to = "value",
               names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
  scale_x_discrete(labels = c("Pre", "Post"))
```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/flexibilidad.png")
```

```
size_effect =
  cohen.d(pre_post$Total_post,
          pre_post$Total_pre, paired = TRUE)

size_effect
```

3.3.4.3.4 Tamaño del efecto

```
##
## Cohen's d
##
## d estimate: 0.067681 (negligible)
## 95 percent confidence interval:
##      lower      upper
## -0.0089023  0.1442636
```

```
# Separamos los descriptivos importantes
```

```
desc = summ[,1:10] %>%
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
```

```

mutate(Prueba = "Flexibilidad") %>%
mutate(Area = "Funciones ejecutivas") %>%
mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Flexibilidad",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)

```

3.4 Socioemocionales

```

#Pre

socioemocional_pre = read_sheet(url_pre, sheet = "Socioemocional")

socioemocional_pre = socioemocional_pre[,1:15]

socioemocional_pre = filter(socioemocional_pre,
  !is.na(`Código`),
  !is.na(`1`),
  `1` != 'NULL')

socioemocional_pre[,7:15] = apply(socioemocional_pre[,7:15], 2,
  function(x) str_to_upper(x))

#Post

socioemocional_post = read_sheet(url_post, sheet = "Socioemocional")

socioemocional_post = socioemocional_post[,1:15]

socioemocional_post = filter(socioemocional_post,
  !is.na(`Código`),
  !is.na(`1`),
  #      !is.na(`Nombre del estudiante`),
  `1` != 'NULL')

socioemocional_post[,7:15] = apply(socioemocional_post[,7:15], 2,
  function(x) str_to_upper(x))

```

```
socioemocional_post$Código = as.character(socioemocional_post$Código)
```

En esta ocasión haremos una base individual para cada constructo pero adicionalmente, obtendremos un alpha general de la prueba

```
#####
# Calificación Reconocimiento

# Pre

socio_reconocimiento_pre = as_tibble(socioemocional_pre) %>%
  dplyr::select(`Código`, "3", "5", "7")

claves_reconocimiento_pre = c("A", "C", "A")

socio_reconocimiento_pre[,2:4] =
  calificacion(data = socio_reconocimiento_pre[,2:4],
              claves = claves_reconocimiento_pre)

socio_reconocimiento_pre[,2:4] =
  as.data.frame(apply(socio_reconocimiento_pre[,2:4],
                    2, function(x) as.numeric(x)))

# Post

socio_reconocimiento_post = as_tibble(socioemocional_post) %>%
  dplyr::select(`Código`, "4", "6", "7")

claves_reconocimiento_post = c("A", "C", "B")

socio_reconocimiento_post[,2:4] =
  calificacion(data = socio_reconocimiento_post[,2:4],
              claves = claves_reconocimiento_post)

socio_reconocimiento_post[,2:4] =
  as.data.frame(apply(socio_reconocimiento_post[,2:4], 2,
                    function(x) as.numeric(x)))

#####
# Calificación Regulación

# pre

regulacion_pre = as_tibble(socioemocional_pre) %>%
  dplyr::select(`Código`, "1", "4", "8")
```

```

regulacion_pre$`1` = str_replace_all(regulacion_pre$`1`,
                                     c("A" = "1", "B" = "1",
                                       "C" = "0", "D" = "0",
                                       "X" = "", "O" = ""))

regulacion_pre$`4` = str_replace_all(regulacion_pre$`4`,
                                     c("A" = "1", "B" = "0",
                                       "C" = "0", "D" = "1",
                                       "X" = "", "O" = ""))

regulacion_pre$`8` = str_replace_all(regulacion_pre$`8`,
                                     c("A" = "1", "B" = "1",
                                       "C" = "0", "D" = "0",
                                       "X" = "", "O" = ""))

regulacion_pre[2:4] = as.data.frame(apply(regulacion_pre[2:4], 2,
                                     function(x) as.numeric(x)))

# post

regulacion_post = as_tibble(socioemocional_post) %>%
  dplyr::select(`Código`, "1", "3", "8")

regulacion_post$`1` = str_replace_all(regulacion_post$`1`,
                                     c("A" = "0", "B" = "0",
                                       "C" = "1", "D" = "1",
                                       "X" = "", "O" = ""))

regulacion_post$`3` = str_replace_all(regulacion_post$`3`,
                                     c("A" = "1", "B" = "1",
                                       "C" = "0", "D" = "0",
                                       "X" = "", "O" = ""))

regulacion_post$`8` = str_replace_all(regulacion_post$`8`,
                                     c("A" = "0", "B" = "1",
                                       "C" = "0", "D" = "1",
                                       "X" = "", "O" = ""))

regulacion_post[2:4] = as.data.frame(apply(regulacion_post[2:4], 2,
                                     function(x) as.numeric(x)))

#####
# Calificación Expresión

```

```

# pre

expresion_pre = as_tibble(socioemocional_pre) %>%
  dplyr::select(`Código`, "2", "6", "9")

expresion_pre$`2` = str_replace_all(expresion_pre$`2`,
  c("A" = "0", "B" = "1",
    "C" = "1", "D" = "0",
    "X" = "", "0" = ""))

expresion_pre$`6` = str_replace_all(expresion_pre$`6`,
  c("A" = "1", "B" = "1",
    "C" = "0", "D" = "0",
    "X" = "", "0" = ""))

expresion_pre$`9` = str_replace_all(expresion_pre$`9`,
  c("A" = "1", "B" = "0",
    "C" = "0", "D" = "1",
    "X" = "", "0" = ""))

expresion_pre[,2:4] = as.data.frame(apply(expresion_pre[,2:4], 2,
  function(x) as.numeric(x)))

# post

expresion_post = as_tibble(socioemocional_post) %>%
  dplyr::select(`Código`, "2", "5", "9")

expresion_post$`2` = str_replace_all(expresion_post$`2`,
  c("A" = "0", "B" = "1",
    "C" = "0", "D" = "1",
    "X" = "", "0" = ""))

expresion_post$`5` = str_replace_all(expresion_post$`5`,
  c("A" = "1", "B" = "0",
    "C" = "0", "D" = "1",
    "X" = "", "0" = ""))

expresion_post$`9` = str_replace_all(expresion_post$`9`,
  c("A" = "1", "B" = "0",
    "C" = "0", "D" = "1",
    "X" = "", "0" = ""))

```

```
expresion_post[,2:4] = as.data.frame(apply(expresion_post[,2:4], 2,
                                           function(x) as.numeric(x)))
```

3.4.1 Alpha

```
alpha_socio_pre = dplyr::select(cbind(regulacion_pre,
                                       socio_reconocimiento_pre,
                                       expresion_pre),
                                -`Código`)

alpha(alpha_socio_pre)
```

3.4.1.1 Total pre

```
## [1] 0.48136
```

```
for(i in seq(length(colnames(alpha_socio_pre)))){
  x = alpha(alpha_socio_pre[, -i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(alpha_socio_pre)[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.434246888323364 al eliminar el ítem 1"
## [1] "El índice de confiabilidad cambia a 0.414461688446607 al eliminar el ítem 4"
## [1] "El índice de confiabilidad cambia a 0.451525382836015 al eliminar el ítem 8"
## [1] "El índice de confiabilidad cambia a 0.463181298843877 al eliminar el ítem 3"
## [1] "El índice de confiabilidad cambia a 0.443300815307036 al eliminar el ítem 5"
## [1] "El índice de confiabilidad cambia a 0.462699141678278 al eliminar el ítem 7"
## [1] "El índice de confiabilidad cambia a 0.436517839692968 al eliminar el ítem 2"
## [1] "El índice de confiabilidad cambia a 0.530677979563551 al eliminar el ítem 6"
## [1] "El índice de confiabilidad cambia a 0.427005253930118 al eliminar el ítem 9"
```

```
alpha_socio_post = dplyr::select(cbind(regulacion_post,
                                       socio_reconocimiento_post,
                                       expresion_post),
                                -`Código`)

alpha(alpha_socio_post)
```

3.4.1.2 Total post

```
## [1] 0.50369
```

```
for(i in seq(length(colnames(alpha_socio_post)))){
  x = alpha(alpha_socio_post[, -i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
```

```

        colnames(alpha_socio_post)[i]))
}

## [1] "El índice de confiabilidad cambia a 0.447831581958777 al eliminar el ítem 1"
## [1] "El índice de confiabilidad cambia a 0.559526083369124 al eliminar el ítem 3"
## [1] "El índice de confiabilidad cambia a 0.467749077181032 al eliminar el ítem 8"
## [1] "El índice de confiabilidad cambia a 0.476850149860897 al eliminar el ítem 4"
## [1] "El índice de confiabilidad cambia a 0.489261344343275 al eliminar el ítem 6"
## [1] "El índice de confiabilidad cambia a 0.477423236811938 al eliminar el ítem 7"
## [1] "El índice de confiabilidad cambia a 0.459220361694422 al eliminar el ítem 2"
## [1] "El índice de confiabilidad cambia a 0.442716655057958 al eliminar el ítem 5"
## [1] "El índice de confiabilidad cambia a 0.442702831071507 al eliminar el ítem 9"

# Amacemos los valores

alfa = rbind(alfa,
              c("Socioemocionales - Pre",
                alpha(alpha_socio_pre)))

alfa = rbind(alfa,
              c("Socioemocionales - Post",
                alpha(alpha_socio_post)))

```

```
alpha(dplyr::select(regulacion_pre, -`Código`))
```

3.4.1.3 Regulación pre

```

## [1] 0.3653

for(i in seq(length(colnames(dplyr::select(regulacion_pre, -`Código`))))){
  x = alpha(dplyr::select(regulacion_pre, -`Código`)[-i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(dplyr::select(regulacion_pre, -`Código`))[i]))
}

## [1] "El índice de confiabilidad cambia a 0.298001954856455 al eliminar el ítem 1"
## [1] "El índice de confiabilidad cambia a 0.178761171032358 al eliminar el ítem 4"
## [1] "El índice de confiabilidad cambia a 0.365562661067099 al eliminar el ítem 8"

```

```
alpha(dplyr::select(regulacion_post, -`Código`))
```

3.4.1.4 Regulación post

```

## [1] 0.077764

for(i in seq(length(colnames(dplyr::select(regulacion_post, -`Código`))))){
  x = alpha(dplyr::select(regulacion_post, -`Código`)[-i])
  print(paste("El índice de confiabilidad cambia a", x,

```



```

        "al eliminar el ítem",
        colnames(dplyr::select(regulacion_post, -`Código`))[i]))
}

```

```

## [1] "El índice de confiabilidad cambia a -0.0265961564316823 al eliminar el ítem 1"
## [1] "El índice de confiabilidad cambia a 0.37341645192649 al eliminar el ítem 3"
## [1] "El índice de confiabilidad cambia a -0.0774420164773457 al eliminar el ítem 8"

```

```
alpha(dplyr::select(expresion_pre, -`Código`))
```

3.4.1.5 Expresión pre

```
## [1] 0.073894
```

```

for(i in seq(length(colnames(dplyr::select(expresion_pre, -`Código`))))){
  x = alpha(dplyr::select(expresion_pre, -`Código`)[-i])
  print(paste("El índice de confiabilidad cambia a", x,
             "al eliminar el ítem",
             colnames(dplyr::select(expresion_pre, -`Código`))[i]))
}

```

```

## [1] "El índice de confiabilidad cambia a -0.0679762212458002 al eliminar el ítem 2"
## [1] "El índice de confiabilidad cambia a 0.191769086642438 al eliminar el ítem 6"
## [1] "El índice de confiabilidad cambia a 0.0383935486540974 al eliminar el ítem 9"

```

```
alpha(dplyr::select(expresion_post, -`Código`))
```

3.4.1.6 Expresión post

```
## [1] 0.52354
```

```

for(i in seq(length(colnames(dplyr::select(expresion_post, -`Código`))))){
  x = alpha(dplyr::select(expresion_post, -`Código`)[-i])
  print(paste("El índice de confiabilidad cambia a", x,
             "al eliminar el ítem",
             colnames(dplyr::select(expresion_post, -`Código`))[i]))
}

```

```

## [1] "El índice de confiabilidad cambia a 0.39609263527604 al eliminar el ítem 2"
## [1] "El índice de confiabilidad cambia a 0.355883611194473 al eliminar el ítem 5"
## [1] "El índice de confiabilidad cambia a 0.535837530769201 al eliminar el ítem 9"

```

```
alpha(dplyr::select(socio_reconocimiento_pre, -`Código`))
```

3.4.1.7 Reconocimiento pre

```
## [1] 0.46758
```

```
for(i in seq(length(colnames(dplyr::select(socio_reconocimiento_pre, -`Código`)))){
  x = alpha(dplyr::select(socio_reconocimiento_pre, -`Código`)[-i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(dplyr::select(socio_reconocimiento_pre, -`Código`))[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.322062421674524 al eliminar el ítem 3"
## [1] "El índice de confiabilidad cambia a 0.345898459970744 al eliminar el ítem 5"
## [1] "El índice de confiabilidad cambia a 0.420927096982472 al eliminar el ítem 7"
```

```
alpha(dplyr::select(socio_reconocimiento_post, -`Código`))
```

3.4.1.8 Reconocimiento post

```
## [1] 0.51621
```

```
for(i in seq(length(colnames(dplyr::select(socio_reconocimiento_post, -`Código`)))){
  x = alpha(dplyr::select(socio_reconocimiento_post, -`Código`)[-i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(dplyr::select(socio_reconocimiento_post, -`Código`))[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.415938862250533 al eliminar el ítem 4"
## [1] "El índice de confiabilidad cambia a 0.405156734781213 al eliminar el ítem 6"
## [1] "El índice de confiabilidad cambia a 0.42455532740377 al eliminar el ítem 7"
```

3.4.2 Indicadores psicométricos

```
analitem = item.exam(alpha_socio_pre, discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
```

```
style = list(
  cell_text(align="center")
),
locations = cells_body())
```

3.4.2.0.1 Total pre

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.261	0.444	0.286	0.926	0.207	0.116
0.367	0.530	0.316	0.840	0.414	0.194
0.423	0.480	0.216	0.767	0.470	0.203
0.458	0.476	0.186	0.700	0.499	0.218
0.321	0.436	0.237	0.883	0.266	0.140
0.311	0.373	0.174	0.892	0.215	0.116
0.323	0.452	0.254	0.882	0.297	0.146
0.420	0.293	0.010	0.229	0.272	0.123
0.431	0.526	0.266	0.754	0.482	0.226

```
# Guardamos los indicadores importantes
```

```
psicometricos_pre = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Socioemocionales - Pre") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_pre)
```

```
analitem = item.exam(alpha_socio_post, discrim=T)
analitem = dplyr::select(analitem, -c("Item.Criterion", "Item.Validity"))
cols_num = colnames(analitem[,1:6])
analitem$Item = row.names(analitem)
analitem[,1:6] %>%
  gt(rowname_col = "Item") %>%
  fmt_number(
    columns = cols_num,
    decimals = 3
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
```

```
style = list(
  cell_text(align="center")
),
locations = cells_body())
```

3.4.2.0.2 Total post

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.294	0.498	0.321	0.904	0.237	0.146
0.416	0.294	0.006	0.222	0.287	0.122
0.241	0.427	0.276	0.938	0.169	0.103
0.442	0.502	0.220	0.735	0.479	0.222
0.416	0.456	0.185	0.778	0.456	0.190
0.354	0.439	0.210	0.853	0.325	0.155
0.263	0.453	0.290	0.925	0.198	0.119
0.313	0.510	0.322	0.890	0.299	0.160
0.405	0.543	0.297	0.793	0.459	0.220

```
# Guardamos los indicadores importantes

psicometricos_post = analitem[,1:6] %>%
  dplyr::select(c("Difficulty", "Discrimination")) %>%
  mutate(Prueba = "Socioemocionales - Post") %>%
  mutate(Item = row.names(.))

indicadores_psicometricos = rbind(indicadores_psicometricos,
                                   psicometricos_post)
```

3.4.3 Comparación pre-post

```
regulacion_pre$Total_pre = apply(regulacion_pre[,2:4], 1,
                                 function (x) sum(x, na.rm = TRUE))

regulacion_post$Total_post = apply(regulacion_post[,2:4], 1,
                                   function (x) sum(x, na.rm = TRUE))

regulacion_pre$Total_pre = min_max_scale(regulacion_pre$Total_pre)
regulacion_post$Total_post = min_max_scale(regulacion_post$Total_post)

pre_post = inner_join(regulacion_post,
                      dplyr::select(regulacion_pre, c("Código",
                                                         "Total_pre")),
                      by = "Código")

pivot_to_bind = dplyr::select(pre_post, c("Código",
```

```

                                Total_pre,
                                Total_post))

pivot_to_bind[,2:ncol(pivot_to_bind)] =
  apply(pivot_to_bind[,2:ncol(pivot_to_bind)],
        2, function(x) 1 - min_max_scale(x))

pivot_to_bind =
  pivot_to_bind %>%
  pivot_longer(cols = !`Código`,
               names_to = c("Prueba", "Tipo"),
               names_sep = "_",
               values_to = "score")

pivot_to_bind$Prueba = str_replace_all(pivot_to_bind$Prueba,
                                       c("Total" = "Socioemocionales - Regulación"))

pivot_final = rbind(pivot_final, pivot_to_bind)

```

3.4.3.1 Regulación

```
print("Estadístico de normalidad pre")
```

3.4.3.1.1 Estadísticos de normalidad

```
## [1] "Estadístico de normalidad pre"
```

```
lillie.test(pre_post$Total_pre)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_pre
## D = 0.371, p-value <0.0000000000000002
```

```
print("Estadístico de normalidad post")
```

```
## [1] "Estadístico de normalidad post"
```

```
lillie.test(pre_post$Total_post)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_post
## D = 0.36, p-value <0.0000000000000002
```

```
summ = data.frame(unclass(psych::describe(pre_post[,c("Total_pre", "Total_post")])))
```

```

        check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Total_pre", "Total_post")

summ[,1:10] %>%
  gt()

```

3.4.3.1.2 Descriptivos

vars	n	mean	sd	median	trimmed	mad	min	max	range
Total_pre	883	0.82333	0.25789	1.00000	0.87364	0	0	1	1
Total_post	883	0.67044	0.21055	0.66667	0.68411	0	0	1	1

```

comparacion =
  wilcox.test(
    x      = pre_post$Total_pre,
    y      = pre_post$Total_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )

comparacion

```

3.4.3.1.3 Comparación de medias

```

##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Total_pre and pre_post$Total_post
## V = 140674, p-value <0.0000000000000002
## alternative hypothesis: true location shift is not equal to 0

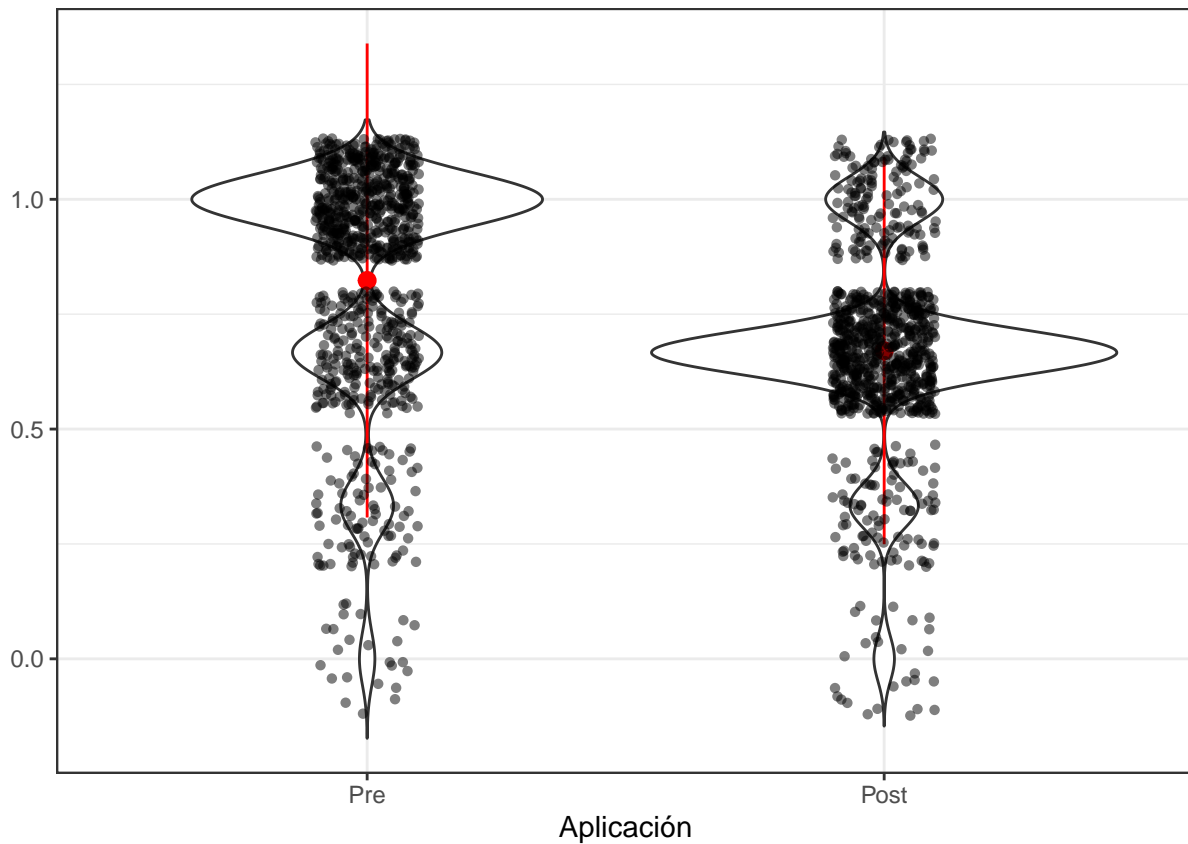
```

```

pre_post %>%
  dplyr::select(Total_pre, Total_post) %>%
  pivot_longer(cols = c(Total_pre, Total_post),
               values_to = "value",
               names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +

```

```
geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
scale_x_discrete(labels = c("Pre", "Post"))
```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/socioemocionales_regulacion.png")
```

```
size_effect =
  cohen.d(pre_post$Total_post,
          pre_post$Total_pre, paired = TRUE)

size_effect
```

3.4.3.1.4 Tamaño del efecto

```
##
## Cohen's d
##
## d estimate: -0.64677 (medium)
## 95 percent confidence interval:
##   lower   upper
## -0.73321 -0.56032
```

```

# Separamos los descriptivos importantes
desc = summ[,1:10] %>%
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
  mutate(Prueba = "Socioemocionales - Regulación") %>%
  mutate(Area = "Habilidades socioemocionales") %>%
  mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Socioemocionales - Regulación",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)

socio_reconocimiento_pre$Total_pre = apply(socio_reconocimiento_pre[,2:4], 1,
  function(x) sum(x, na.rm= FALSE))

socio_reconocimiento_post$Total_post = apply(socio_reconocimiento_post[,2:4], 1,
  function(x) sum(x, na.rm= FALSE))

socio_reconocimiento_pre$Total_pre =
  min_max_scale(socio_reconocimiento_pre$Total_pre)

socio_reconocimiento_post$Total_post =
  min_max_scale(socio_reconocimiento_post$Total_post)

pre_post = inner_join(socio_reconocimiento_post,
  dplyr::select(socio_reconocimiento_pre, c("Código",
    "Total_pre")),
  by = "Código")

pivot_to_bind = dplyr::select(pre_post, c("Código",
  Total_pre,
  Total_post))

pivot_to_bind[,2:ncol(pivot_to_bind)] =
  apply(pivot_to_bind[,2:ncol(pivot_to_bind)],
    2, function(x) 1 - min_max_scale(x))

```



```

pivot_to_bind =
  pivot_to_bind %>%
  pivot_longer(cols = !`Código`,
               names_to = c("Prueba", "Tipo"),
               names_sep = "_",
               values_to = "score")

pivot_to_bind$Prueba = str_replace_all(pivot_to_bind$Prueba,
                                       c("Total" = "Socioemocionales - Reconocimiento"))

pivot_final = rbind(pivot_final, pivot_to_bind)

```

3.4.3.2 Reconocimiento

```
print("Estadístico de normalidad pre")
```

3.4.3.2.1 Estadísticos de normalidad

```
## [1] "Estadístico de normalidad pre"
```

```
lillie.test(pre_post$Total_pre)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_pre
## D = 0.358, p-value <0.0000000000000002
```

```
print("Estadístico de normalidad post")
```

```
## [1] "Estadístico de normalidad post"
```

```
lillie.test(pre_post$Total_post)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_post
## D = 0.324, p-value <0.0000000000000002
```

```

summ = data.frame(unclass(psych::describe(pre_post[,c("Total_pre",
                                                       "Total_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Total_pre", "Total_post")

summ[,1:10] %>%
  gt()

```

3.4.3.2.2 Descriptivos

vars	n	mean	sd	median	trimmed	mad	min	max	range
Total_pre	883	0.81049	0.26432	1	0.85809	0	0	1	1
Total_post	883	0.76821	0.30055	1	0.81848	0	0	1	1

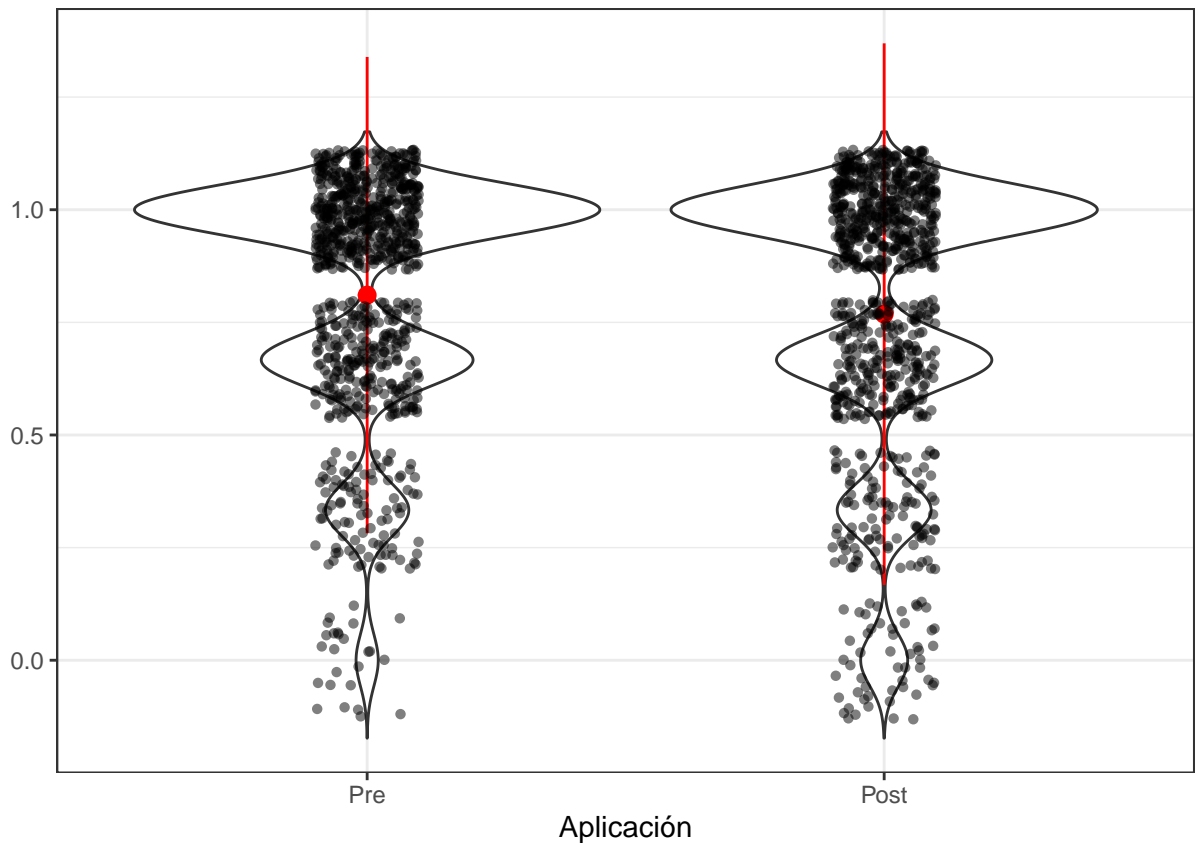
```
comparacion =  
  wilcox.test(  
    x      = pre_post$Total_pre,  
    y      = pre_post$Total_post,  
    alternative = "two.sided",  
    mu      = 0,  
    var.equal = TRUE,  
    paired   = TRUE,  
    conf.level = 0.95  
  )
```

```
comparacion
```

3.4.3.2.3 Comparación de medias

```
##  
## Wilcoxon signed rank test with continuity correction  
##  
## data: pre_post$Total_pre and pre_post$Total_post  
## V = 53868, p-value = 0.0011  
## alternative hypothesis: true location shift is not equal to 0
```

```
pre_post %>%  
  dplyr::select(Total_pre, Total_post) %>%  
  pivot_longer(cols = c(Total_pre, Total_post),  
               values_to = "value",  
               names_to = "Aplicación") %>%  
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +  
  geom_violin(trim=FALSE) +  
  theme_bw() +  
  theme(legend.position = "none") +  
  xlab("Aplicación") + ylab("") +  
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +  
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +  
  scale_x_discrete(labels = c("Pre", "Post"))
```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/socioemocionales_reconocimiento.png")
```

```
size_effect =
  cohen.d(pre_post$Total_post,
          pre_post$Total_pre, paired = TRUE)

size_effect
```

3.4.3.2.4 Tamaño del efecto

```
##
## Cohen's d
##
## d estimate: -0.1491 (negligible)
## 95 percent confidence interval:
##   lower      upper
## -0.226490 -0.071716
```

```
# Separamos los descriptivos importantes
```

```
desc = summ[,1:10] %>%
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
```

```

mutate(Prueba = "Socioemocionales - Reconocimiento") %>%
mutate(Area = "Habilidades socioemocionales") %>%
mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Socioemocionales - Reconocimiento",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)

expresion_pre$Total_pre = apply(expresion_pre[,2:4], 1,
  function (x) sum(x, na.rm = TRUE))

expresion_post$Total_post = apply(expresion_post[,2:4], 1,
  function (x) sum(x, na.rm = TRUE))

expresion_pre$Total_pre = min_max_scale(expresion_pre$Total_pre)
expresion_post$Total_post = min_max_scale(expresion_post$Total_post)

pre_post = inner_join(expresion_post,
  dplyr::select(expresion_pre, c("Código", "Total_pre")),
  by = "Código")

pivot_to_bind = dplyr::select(pre_post, c("Código",
  Total_pre,
  Total_post))

pivot_to_bind[,2:ncol(pivot_to_bind)] =
  apply(pivot_to_bind[,2:ncol(pivot_to_bind)],
    2, function(x) 1 - min_max_scale(x))

pivot_to_bind =
  pivot_to_bind %>%
  pivot_longer(cols = !`Código`,
    names_to = c("Prueba", "Tipo"),
    names_sep = "_",
    values_to = "score")

```

```

pivot_to_bind$Prueba = str_replace_all(pivot_to_bind$Prueba,
                                         c("Total" = "Socioemocionales - Expresión"))

pivot_final = rbind(pivot_final, pivot_to_bind)

```

3.4.3.3 Expresión

```

print("Estadístico de normalidad pre")

```

3.4.3.3.1 Estadísticos de normalidad

```

## [1] "Estadístico de normalidad pre"

```

```

lillie.test(pre_post$Total_pre)

```

```

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_pre
## D = 0.317, p-value <0.0000000000000002

```

```

print("Estadístico de normalidad post")

```

```

## [1] "Estadístico de normalidad post"

```

```

lillie.test(pre_post$Total_post)

```

```

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$Total_post
## D = 0.413, p-value <0.0000000000000002

```

```

summ = data.frame(unclass(psych::describe(pre_post[,c("Total_pre", "Total_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("Total_pre", "Total_post")

summ[,1:10] %>%
  gt()

```

3.4.3.3.2 Descriptivos

vars	n	mean	sd	median	trimmed	mad	min	max	range
Total_pre	883	0.61382	0.24224	0.66667	0.61858	0	0	1	1
Total_post	883	0.85089	0.26138	1.00000	0.91278	0	0	1	1

```

comparacion =
  wilcox.test(
    x      = pre_post$Total_pre,
    y      = pre_post$Total_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )

```

```
comparacion
```

3.4.3.3.3 Comparación de medias

```

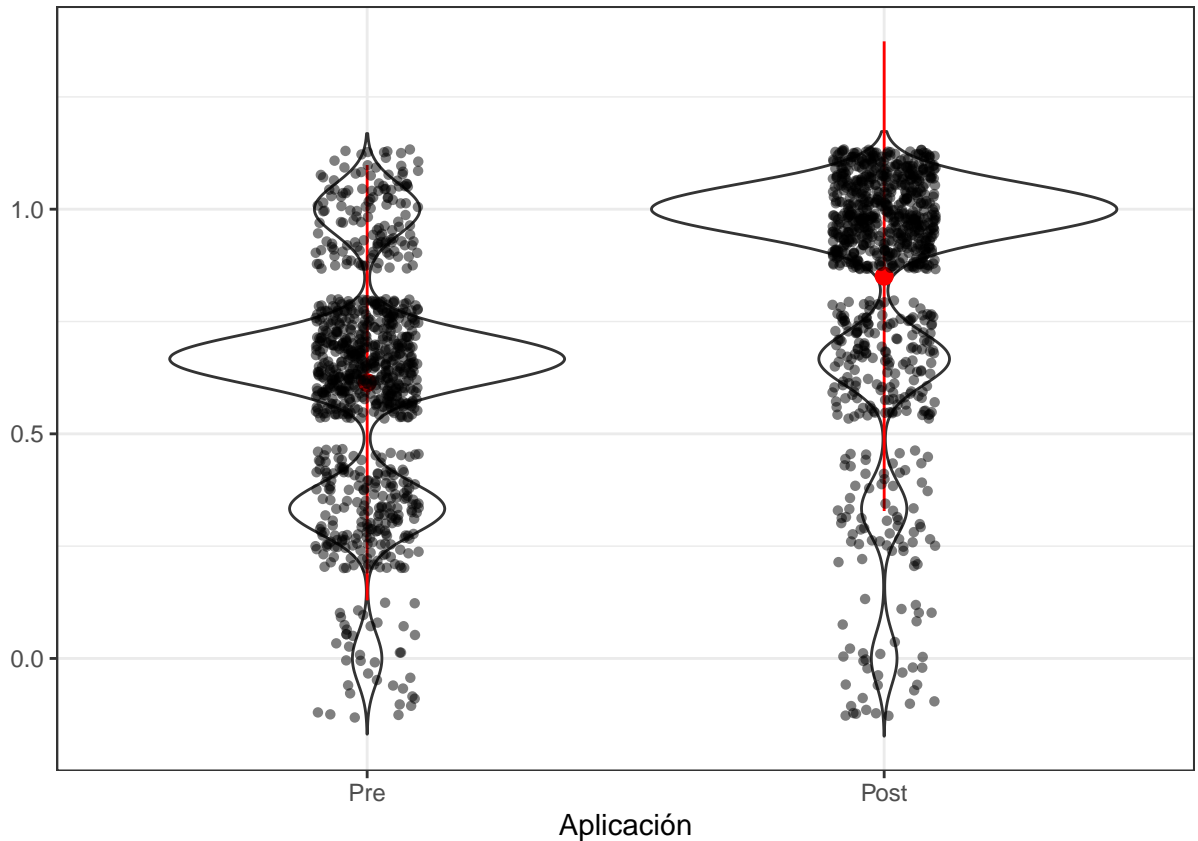
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Total_pre and pre_post$Total_post
## V = 24476, p-value <0.0000000000000002
## alternative hypothesis: true location shift is not equal to 0

```

```

pre_post %>%
  dplyr::select(Total_pre, Total_post) %>%
  pivot_longer(cols = c(Total_pre, Total_post),
               values_to = "value",
               names_to = "Aplicación") %>%
  ggplot(aes(x = reorder(`Aplicación`, desc(`Aplicación`)), y = value)) +
  geom_violin(trim=FALSE) +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Aplicación") + ylab("") +
  stat_summary(fun.data=mean_sdl, geom="pointrange", color="red") +
  geom_jitter(shape=16, position=position_jitter(0.1), alpha=0.5) +
  scale_x_discrete(labels = c("Pre", "Post"))

```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/socioemocionales_expresion.png")
```

```
size_effect =
  cohen.d(pre_post$Total_post,
          pre_post$Total_pre, paired = TRUE)

size_effect
```

3.4.3.3.4 Tamaño del efecto

```
##
## Cohen's d
##
## d estimate: 0.94061 (large)
## 95 percent confidence interval:
##   lower   upper
## 0.83557 1.04564
```

```
# Separamos los descriptivos importantes
```

```
desc = summ[,1:10] %>%
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
```

```

mutate(Prueba = "Socioemocionales - Expresión") %>%
mutate(Area = "Habilidades socioemocionales") %>%
mutate(`Aplicación` = c("Pre", "Post"))

descriptivos = rbind(descriptivos, desc)

# Guardamos los resultados de esta comparación

data_temp =
  data.frame(Prueba = "Socioemocionales - Expresión",
    `Media pre` = summ$mean[1],
    `Media post` = summ$mean[2],
    `p value` = comparacion$p.value,
    `D de cohen` = size_effect$estimate)

comparaciones = rbind(comparaciones, data_temp)

```


4 Resúmenes de datos

```
library(scales)
```

4.1 Alpha de las pruebas

```
alfa$Alfa = as.numeric(alfa$Alfa)
alfa$Prueba = chartr("áéíóú", "aeiou", alfa$Prueba)

drop_na(alfa) %>%
  gt() %>%
  fmt_number(
    columns = "Alfa",
    decimals = 2
  ) %>%
  data_color(columns = "Alfa",
             colors = col_numeric(palette = c("red", "green"),
                                   domain = c(0,1))) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body()
```

Prueba	Alfa
Actitudes lenguaje pre	0.78
Actitudes lenguaje post	0.85
Actitudes matematicas pre	0.80
Actitudes matematicas post	0.75
Motivacion pre	0.29
Motivacion post	0.66
Memoria auditiva (Item 1) - Pre	0.68
Memoria auditiva (Item 2) - Pre	0.64
Memoria auditiva (Item 1) - Post	0.69
Memoria auditiva (Item 2) - Post	0.59
Memoria visual (Item 1) - Pre	0.47
Memoria visual (Item 2) - Pre	0.43
Memoria visual (Item 1) - Post	0.44
Memoria visual (Item 2) - Post	0.35
Inhibicion - Pre	0.93

Inhibicion - Post	0.94
Flexibilidad - Pre	0.91
Flexibilidad - Post	0.92
Socioemocionales - Pre	0.48
Socioemocionales - Post	0.50

```
# Exporte
```

```
write.xlsx(drop_na(alfa), "../Data/processed/segunda_aplicacion/alfa.xlsx", row.names = FALSE)
```

4.2 Descriptivos

```
descriptivos$Prueba = chartr("áéíóú", "aeiou", descriptivos$Prueba)
colnames(descriptivos) = chartr("áéíóú", "aeiou", colnames(descriptivos))
```

```
dplyr::select(descriptivos, -Area) %>%
  gt() %>%
  fmt_number(
    columns = c('mean', 'sd'),
    decimals = 2
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body()
```

n	mean	sd	min	max	Prueba	Aplicacion
433	0.78	0.21	0	1	Actitudes hacia el lenguaje - Total	Pre
433	0.86	0.21	0	1	Actitudes hacia el lenguaje - Total	Post
433	0.80	0.26	0	1	Actitudes hacia el lenguaje - Afectivo	Pre
433	0.86	0.23	0	1	Actitudes hacia el lenguaje - Afectivo	Post
433	0.85	0.23	0	1	Actitudes hacia el lenguaje - Cognitivo	Pre
433	0.87	0.22	0	1	Actitudes hacia el lenguaje - Cognitivo	Post
433	0.69	0.25	0	1	Actitudes hacia el lenguaje - Conativo	Pre
433	0.84	0.25	0	1	Actitudes hacia el lenguaje - Conativo	Post
452	0.89	0.18	0	1	Actitudes hacia las matematicas - Total	Pre
452	0.91	0.17	0	1	Actitudes hacia las matematicas - Total	Post
452	0.83	0.27	0	1	Actitudes hacia las matematicas - Afectivo	Pre

452	0.90	0.21	0	1	Actitudes hacia las matematicas - Afectivo	Post
452	0.91	0.19	0	1	Actitudes hacia las matematicas - Cognitivo	Pre
452	0.93	0.18	0	1	Actitudes hacia las matematicas - Cognitivo	Post
452	0.91	0.19	0	1	Actitudes hacia las matematicas - Conativo	Pre
452	0.91	0.20	0	1	Actitudes hacia las matematicas - Conativo	Post
887	0.68	0.14	0	1	Motivacion - Total	Pre
887	0.79	0.19	0	1	Motivacion - Total	Post
887	0.62	0.16	0	1	Motivacion - Interes	Pre
887	0.88	0.25	0	1	Motivacion - Interes	Post
887	0.62	0.22	0	1	Motivacion - Metas	Pre
887	0.70	0.30	0	1	Motivacion - Metas	Post
887	0.66	0.32	0	1	Motivacion - Atribucion interna	Pre
887	0.71	0.33	0	1	Motivacion - Atribucion interna	Post
887	0.86	0.27	0	1	Motivacion - Expectativas	Pre
887	0.86	0.29	0	1	Motivacion - Expectativas	Post
886	0.64	0.22	0	1	Memoria auditiva	Pre
886	0.67	0.21	0	1	Memoria auditiva	Post
887	0.58	0.32	0	1	Memoria visual	Pre
887	0.69	0.30	0	1	Memoria visual	Post
888	0.83	0.30	0	1	Inhibicion	Pre
888	0.89	0.25	0	1	Inhibicion	Post
887	0.85	0.26	0	1	Flexibilidad	Pre
887	0.87	0.26	0	1	Flexibilidad	Post
883	0.82	0.26	0	1	Socioemocionales - Regulacion	Pre
883	0.67	0.21	0	1	Socioemocionales - Regulacion	Post
883	0.81	0.26	0	1	Socioemocionales - Reconocimiento	Pre
883	0.77	0.30	0	1	Socioemocionales - Reconocimiento	Post
883	0.61	0.24	0	1	Socioemocionales - Expresion	Pre
883	0.85	0.26	0	1	Socioemocionales - Expresion	Post

```
# Exporte
```

```
write.xlsx(descriptivos, "../Data/processed/segunda_aplicacion/descriptivos.xlsx", row.names =
```

4.3 Indicadores psicométricos

```
indicadores_psicometricos$Prueba = chartr("áéíóú", "aeiou",
                                             indicadores_psicometricos$Prueba)

indicadores_psicometricos$Item = chartr("áéíóú", "aeiou",
                                           indicadores_psicometricos$Item)

indicadores_psicometricos %>%
  gt() %>%
  fmt_number(
    columns = c("Difficulty", "Discrimination"),
    decimals = 2
```

```

) %>%
data_color(columns = "Difficulty",
           colors = col_numeric(palette = c("white", "#5FA14A"),
                               domain = c(0,1))) %>%
data_color(columns = "Discrimination",
           colors = col_numeric(palette = c("white", "#5FA14A"),
                               domain = c(0,0.3),
                               na.color = "#5FA14A"))) %>%

tab_style(
  style = list(
    cell_text(align="center", weight="bold")
  ),
  locations=cells_column_labels()
)%>%
tab_style(
  style = list(
    cell_text(align="center")
  ),
  locations = cells_body())

```

Difficulty	Discrimination	Prueba	Item
0.82	0.41	Actitudes hacia el lenguaje - Pre	Grupo 1 __ 1
0.89	0.29	Actitudes hacia el lenguaje - Pre	Grupo 1 __ 2
0.81	0.41	Actitudes hacia el lenguaje - Pre	Grupo 1 __ 3
0.72	0.58	Actitudes hacia el lenguaje - Pre	Grupo 2 __ 1
0.85	0.40	Actitudes hacia el lenguaje - Pre	Grupo 2 __ 2
0.86	0.38	Actitudes hacia el lenguaje - Pre	Grupo 2 __ 3
0.92	0.21	Actitudes hacia el lenguaje - Pre	Grupo 3 __ 1
0.91	0.25	Actitudes hacia el lenguaje - Pre	Grupo 3 __ 2
0.32	0.17	Actitudes hacia el lenguaje - Pre	Grupo 3 __ 3
0.72	0.64	Actitudes hacia el lenguaje - Pre	Grupo 4 __ 1
0.76	0.59	Actitudes hacia el lenguaje - Pre	Grupo 4 __ 2
0.77	0.58	Actitudes hacia el lenguaje - Pre	Grupo 4 __ 3
0.91	0.27	Actitudes hacia el lenguaje - Post	Grupo 1 __ 1
0.90	0.29	Actitudes hacia el lenguaje - Post	Grupo 1 __ 2
0.82	0.42	Actitudes hacia el lenguaje - Post	Grupo 1 __ 3
0.87	0.34	Actitudes hacia el lenguaje - Post	Grupo 2 __ 1
0.90	0.28	Actitudes hacia el lenguaje - Post	Grupo 2 __ 2
0.86	0.37	Actitudes hacia el lenguaje - Post	Grupo 2 __ 3
0.76	0.60	Actitudes hacia el lenguaje - Post	Grupo 3 __ 1
0.76	0.64	Actitudes hacia el lenguaje - Post	Grupo 3 __ 2
0.77	0.61	Actitudes hacia el lenguaje - Post	Grupo 3 __ 3
0.85	0.42	Actitudes hacia el lenguaje - Post	Grupo 4 __ 1
0.89	0.30	Actitudes hacia el lenguaje - Post	Grupo 4 __ 2
0.90	0.28	Actitudes hacia el lenguaje - Post	Grupo 4 __ 3
0.86	0.38	Actitudes hacia las matematicas - Pre	Grupo 1 __ 1
0.91	0.23	Actitudes hacia las matematicas - Pre	Grupo 1 __ 2

0.87	0.37	Actitudes hacia las matematicas - Pre	Grupo 1 _ 3
0.74	0.61	Actitudes hacia las matematicas - Pre	Grupo 2 _ 1
0.90	0.29	Actitudes hacia las matematicas - Pre	Grupo 2 _ 2
0.89	0.32	Actitudes hacia las matematicas - Pre	Grupo 2 _ 3
0.93	0.19	Actitudes hacia las matematicas - Pre	Grupo 3 _ 1
0.93	0.20	Actitudes hacia las matematicas - Pre	Grupo 3 _ 2
0.94	0.16	Actitudes hacia las matematicas - Pre	Grupo 3 _ 3
0.87	0.34	Actitudes hacia las matematicas - Pre	Grupo 4 _ 1
0.89	0.32	Actitudes hacia las matematicas - Pre	Grupo 4 _ 2
0.92	0.21	Actitudes hacia las matematicas - Pre	Grupo 4 _ 3
0.87	0.40	Actitudes hacia las matematicas - Post	Grupo 1 _ 1
0.92	0.24	Actitudes hacia las matematicas - Post	Grupo 1 _ 2
0.87	0.38	Actitudes hacia las matematicas - Post	Grupo 1 _ 3
0.93	0.20	Actitudes hacia las matematicas - Post	Grupo 2 _ 1
0.94	0.19	Actitudes hacia las matematicas - Post	Grupo 2 _ 2
0.91	0.28	Actitudes hacia las matematicas - Post	Grupo 2 _ 3
0.11	-0.13	Actitudes hacia las matematicas - Post	Grupo 3 _ 1
0.91	0.25	Actitudes hacia las matematicas - Post	Grupo 3 _ 2
0.90	0.30	Actitudes hacia las matematicas - Post	Grupo 3 _ 3
0.88	0.31	Actitudes hacia las matematicas - Post	Grupo 4 _ 1
0.94	0.18	Actitudes hacia las matematicas - Post	Grupo 4 _ 2
0.91	0.26	Actitudes hacia las matematicas - Post	Grupo 4 _ 3
0.87	0.29	Motivacion - Pre	1
0.89	0.27	Motivacion - Pre	2
0.82	0.39	Motivacion - Pre	3
0.85	0.32	Motivacion - Pre	4
0.91	0.22	Motivacion - Pre	5
0.81	0.38	Motivacion - Pre	6
0.68	0.55	Motivacion - Pre	7
0.05	0.01	Motivacion - Pre	8
0.24	0.01	Motivacion - Pre	9
0.63	0.45	Motivacion - Pre	10
0.86	0.33	Motivacion - Post	1
0.75	0.55	Motivacion - Post	2
0.74	0.44	Motivacion - Post	3
0.87	0.31	Motivacion - Post	4
0.76	0.43	Motivacion - Post	5
0.85	0.33	Motivacion - Post	6
0.88	0.31	Motivacion - Post	7
0.84	0.37	Motivacion - Post	8
0.57	0.57	Motivacion - Post	9
0.64	0.41	Motivacion - Post	10
0.83	0.39	Memoria auditiva (Item 1) - Pre	Caballo
0.71	0.53	Memoria auditiva (Item 1) - Pre	Perro
0.73	0.53	Memoria auditiva (Item 1) - Pre	Aguila
0.80	0.46	Memoria auditiva (Item 1) - Pre	Pollito
0.66	0.59	Memoria auditiva (Item 1) - Pre	Foca
0.70	0.51	Memoria auditiva (Item 1) - Pre	Vaca

0.75	0.49	Memoria auditiva (Item 1) - Pre	Cocodrilo
0.62	0.59	Memoria auditiva (Item 1) - Pre	Sapo
0.71	0.49	Memoria auditiva (Item 2) - Pre	Oveja
0.57	0.64	Memoria auditiva (Item 2) - Pre	Elefante
0.75	0.45	Memoria auditiva (Item 2) - Pre	Tiburón
0.68	0.56	Memoria auditiva (Item 2) - Pre	Caracol
0.72	0.50	Memoria auditiva (Item 2) - Pre	Ratón
0.53	0.58	Memoria auditiva (Item 2) - Pre	Gato
0.73	0.48	Memoria auditiva (Item 2) - Pre	Tortuga
0.76	0.42	Memoria auditiva (Item 2) - Pre	Pez
0.84	0.36	Memoria auditiva (Item 1) - Post	Caballo
0.74	0.49	Memoria auditiva (Item 1) - Post	Gallina
0.77	0.49	Memoria auditiva (Item 1) - Post	Conejo
0.68	0.54	Memoria auditiva (Item 1) - Post	Leopardo
0.78	0.45	Memoria auditiva (Item 1) - Post	Serpiente
0.68	0.57	Memoria auditiva (Item 1) - Post	Cerdo
0.66	0.64	Memoria auditiva (Item 1) - Post	Cangrejo
0.70	0.47	Memoria auditiva (Item 1) - Post	León
0.86	0.33	Memoria auditiva (Item 2) - Post	Ballena
0.50	0.45	Memoria auditiva (Item 2) - Post	Gato
0.55	0.65	Memoria auditiva (Item 2) - Post	Abeja
0.60	0.59	Memoria auditiva (Item 2) - Post	Cebra
0.64	0.59	Memoria auditiva (Item 2) - Post	Rana
0.79	0.43	Memoria auditiva (Item 2) - Post	Cocodrilo
0.69	0.48	Memoria auditiva (Item 2) - Post	Tortuga
0.79	0.38	Memoria auditiva (Item 2) - Post	Iguana
0.75	0.62	Memoria visual (Item 1) - Pre	OFICINA
0.79	0.56	Memoria visual (Item 1) - Pre	LAVANDERIA
0.87	0.37	Memoria visual (Item 1) - Pre	CINE
0.75	0.57	Memoria visual (Item 2) - Pre	BARCO
0.63	0.78	Memoria visual (Item 2) - Pre	JUGUETES
0.71	0.65	Memoria visual (Item 2) - Pre	ELECTRONICOS
0.95	0.16	Memoria visual (Item 1) - Post	Campo
0.80	0.59	Memoria visual (Item 1) - Post	Colegio__
0.87	0.37	Memoria visual (Item 1) - Post	Noira
0.84	0.40	Memoria visual (Item 2) - Post	Heladeria
0.77	0.56	Memoria visual (Item 2) - Post	Parque
0.80	0.52	Memoria visual (Item 2) - Post	Aeropuerto
0.84	0.45	Inhibicion - Pre	Item 2 __ 1
0.83	0.49	Inhibicion - Pre	Item 2 __ 2
0.82	0.54	Inhibicion - Pre	Item 2 __ 3
0.81	0.54	Inhibicion - Pre	Item 2 __ 4
0.81	0.56	Inhibicion - Pre	Item 2 __ 5
0.83	0.51	Inhibicion - Pre	Item 2 __ 6
0.80	0.55	Inhibicion - Pre	Item 2 __ 7
0.82	0.52	Inhibicion - Pre	Item 2 __ 8
0.80	0.57	Inhibicion - Pre	Item 2 __ 9
0.82	0.52	Inhibicion - Pre	Item 2 __ 10

0.92	0.24	Inhibicion - Post	Item 2 __ 1
0.89	0.32	Inhibicion - Post	Item 2 __ 2
0.87	0.40	Inhibicion - Post	Item 2 __ 3
0.87	0.38	Inhibicion - Post	Item 2 __ 4
0.88	0.36	Inhibicion - Post	Item 2 __ 5
0.88	0.36	Inhibicion - Post	Item 2 __ 6
0.87	0.39	Inhibicion - Post	Item 2 __ 7
0.88	0.37	Inhibicion - Post	Item 2 __ 8
0.88	0.35	Inhibicion - Post	Item 2 __ 9
0.86	0.41	Inhibicion - Post	Item 2 __ 10
0.84	0.49	Flexibilidad - Pre	Item 2 __ 1
0.88	0.36	Flexibilidad - Pre	Item 2 __ 2
0.79	0.61	Flexibilidad - Pre	Item 2 __ 3
0.80	0.57	Flexibilidad - Pre	Item 2 __ 4
0.88	0.35	Flexibilidad - Pre	Item 2 __ 5
0.83	0.49	Flexibilidad - Pre	Item 2 __ 6
0.82	0.52	Flexibilidad - Pre	Item 2 __ 7
0.80	0.57	Flexibilidad - Pre	Item 2 __ 8
0.90	0.30	Flexibilidad - Pre	Item 2 __ 9
0.91	0.26	Flexibilidad - Pre	Item 2 __ 10
0.85	0.45	Flexibilidad - Post	Item 2 __ 1
0.90	0.29	Flexibilidad - Post	Item 2 __ 2
0.87	0.39	Flexibilidad - Post	Item 2 __ 3
0.84	0.48	Flexibilidad - Post	Item 2 __ 4
0.88	0.37	Flexibilidad - Post	Item 2 __ 5
0.84	0.47	Flexibilidad - Post	Item 2 __ 6
0.83	0.50	Flexibilidad - Post	Item 2 __ 7
0.87	0.39	Flexibilidad - Post	Item 2 __ 8
0.84	0.47	Flexibilidad - Post	Item 2 __ 9
0.85	0.45	Flexibilidad - Post	Item 2 __ 10
0.93	0.21	Socioemocionales - Pre	1
0.84	0.41	Socioemocionales - Pre	4
0.77	0.47	Socioemocionales - Pre	8
0.70	0.50	Socioemocionales - Pre	3
0.88	0.27	Socioemocionales - Pre	5
0.89	0.22	Socioemocionales - Pre	7
0.88	0.30	Socioemocionales - Pre	2
0.23	0.27	Socioemocionales - Pre	6
0.75	0.48	Socioemocionales - Pre	9
0.90	0.24	Socioemocionales - Post	1
0.22	0.29	Socioemocionales - Post	3
0.94	0.17	Socioemocionales - Post	8
0.73	0.48	Socioemocionales - Post	4
0.78	0.46	Socioemocionales - Post	6
0.85	0.33	Socioemocionales - Post	7
0.93	0.20	Socioemocionales - Post	2
0.89	0.30	Socioemocionales - Post	5

Exporte

```
write.xlsx(indicadores_psicometricos,
           "../Data/processed/segunda_aplicacion/indicadores_psicometricos.xlsx",
           row.names = FALSE)
```

4.4 Pruebas pre-post

```
comparaciones$Prueba = chartr("áéíóú", "aeiou",
                              comparaciones$Prueba)

comparaciones %>%
  gt() %>%
  fmt_number(
    columns = "p.value",
    decimals = 2
  ) %>%
  data_color(columns = "p.value",
             colors = col_numeric(palette = c("#5FA14A", "#5FA14A"),
                                  domain = c(0, 0.05),
                                  na.color = "white")) %>%
  tab_style(
    style = list(
      cell_text(align="center", weight="bold")
    ),
    locations=cells_column_labels()
  ) %>%
  tab_style(
    style = list(
      cell_text(align="center")
    ),
    locations = cells_body()
```

Prueba	Media.pre	Media.post	p.value	D.de.cohen
Actitudes hacia el lenguaje - Total	0.77868	0.85758	0.00	0.3726915
Actitudes hacia el lenguaje - Afectivo	0.79561	0.85566	0.00	0.2395837
Actitudes hacia el lenguaje - Cognitivo	0.85277	0.87298	0.08	0.0892848
Actitudes hacia el lenguaje - Conativo	0.68764	0.84411	0.00	0.6375192
Actitudes hacia las matematicas - Total	0.88576	0.91372	0.00	0.1609706
Actitudes hacia las matematicas - Afectivo	0.82743	0.90487	0.00	0.3199345
Actitudes hacia las matematicas - Cognitivo	0.90819	0.92865	0.04	0.1100097
Actitudes hacia las matematicas - Conativo	0.90708	0.90542	0.70	-0.0084821
Motivacion - Total	0.67610	0.78805	0.00	0.6522443
Motivacion - Interes	0.61969	0.87862	0.00	1.2117739

Motivacion - Metas	0.62420	0.70500	0.00	0.3070580
Motivacion - Atribucion interna	0.65614	0.70857	0.00	0.1615737
Motivacion - Expectativas	0.85851	0.85626	0.78	-0.0081094
Memoria auditiva	0.64285	0.67128	0.00	0.1311863
Memoria visual	0.58437	0.68828	0.00	0.3330531
Inhibicion	0.82703	0.89088	0.00	0.2297209
Flexibilidad	0.85051	0.86809	0.05	0.0676806
Socioemocionales - Regulacion	0.82333	0.67044	0.00	-0.6467682
Socioemocionales - Reconocimiento	0.81049	0.76821	0.00	-0.1491027
Socioemocionales - Expresion	0.61382	0.85089	0.00	0.9406075

Exporte

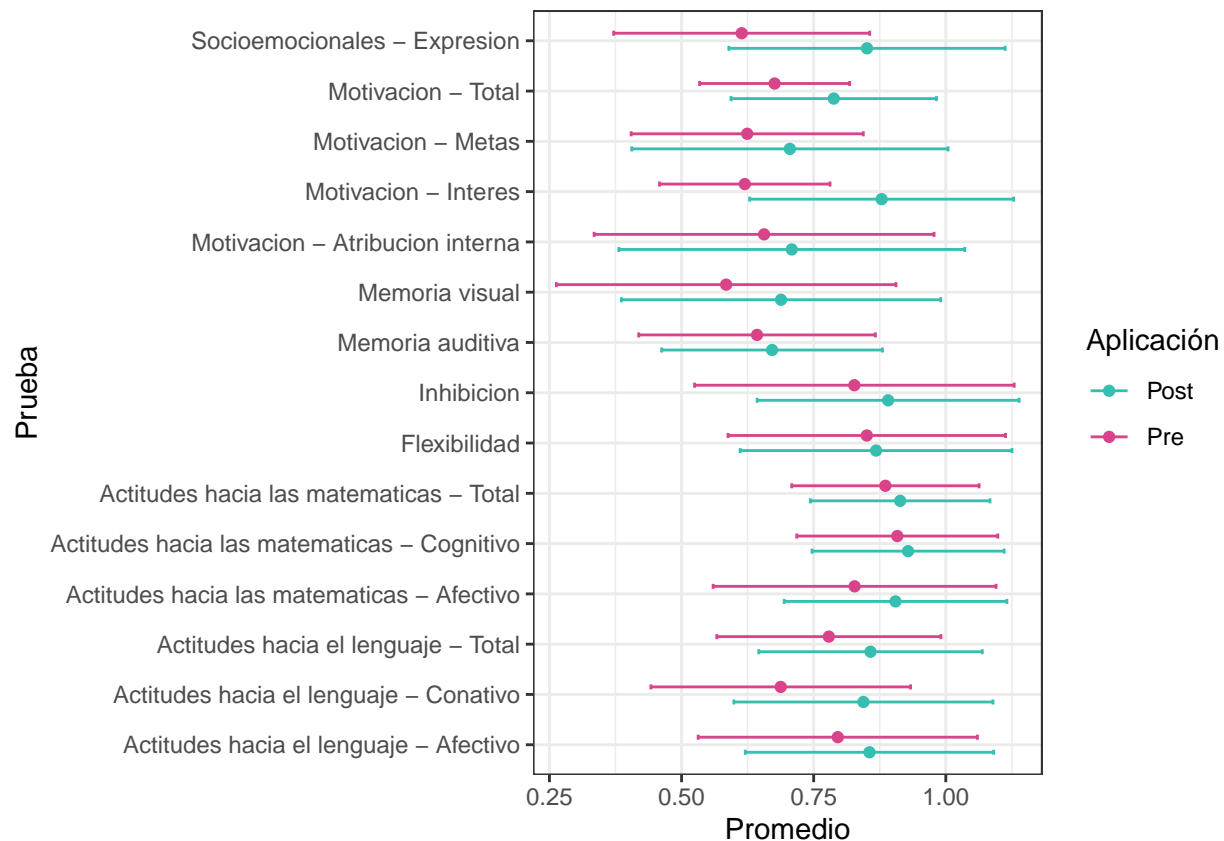
```
write.xlsx(comparaciones,
           "../Data/processed/segunda_aplicacion/pruebas_pre_post.xlsx",
           row.names = FALSE)
```

```
filtro = filter(comparaciones, p.value < 0.05,
                 Media.pre < Media.post)$Prueba
```

```
significativos = descriptivos %>%
  filter(Prueba %in% filtro)
```

```
significativos %>%
  dplyr::select("mean" , "sd", "Prueba", "Aplicacion") %>%
  ggplot(aes(x = Prueba, y = mean, color = Aplicacion)) +
  geom_errorbar(aes(ymin= mean - sd, ymax = mean + sd), width = .2, position=position_dodge(0.6)) +
  geom_point(position=position_dodge(0.6)) +
  scale_color_manual(name = "Aplicación",
                    values=c(colores[2], colores[1]),
                    ) + # Show dots

  xlab("Prueba") +
  ylab("Promedio") +
  coord_flip() +
  labs(fill = "Type") +
  theme_bw()
```



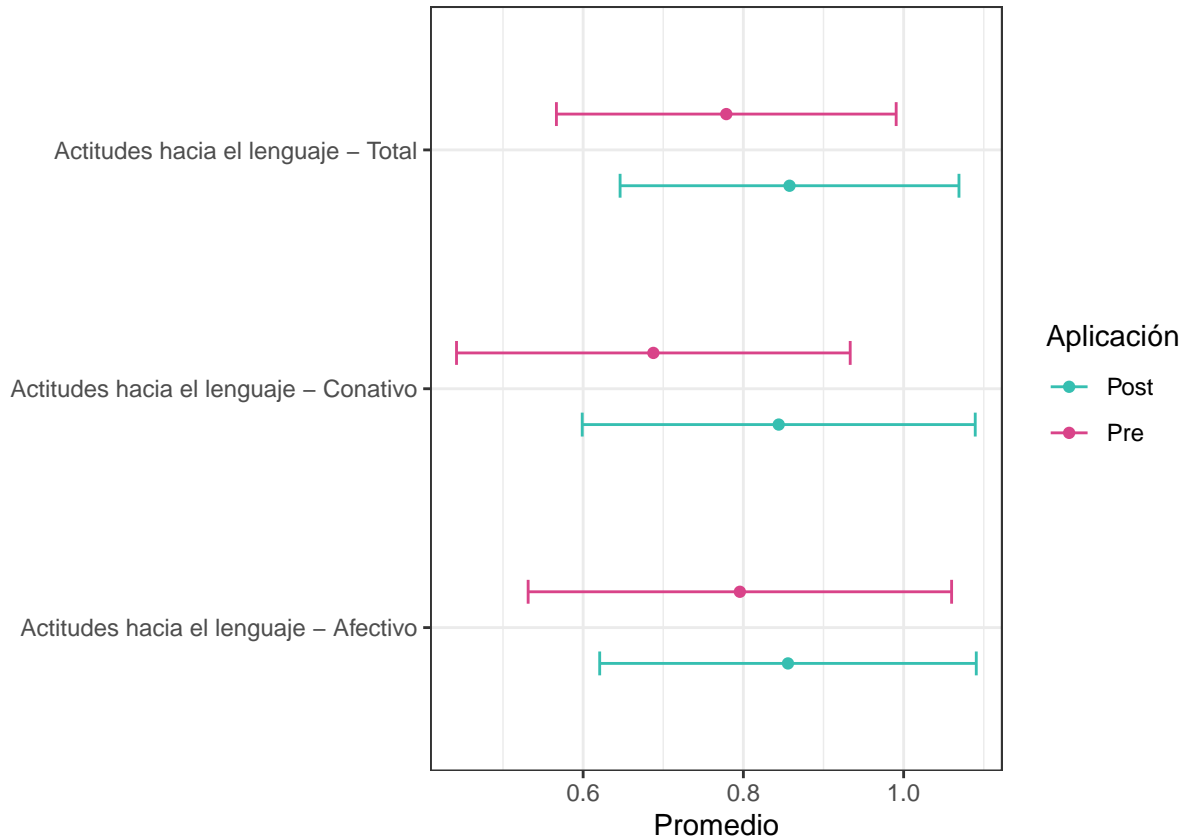
```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/pruebas_significativas.png")
```

4.4.1 Actitudes hacia el lenguaje

```
significativos %>%
  filter(Area == "Actitudes hacia el lenguaje") %>%
  dplyr::select("mean", "sd", "Prueba", "Aplicacion") %>%
  ggplot(aes(x = Prueba, y = mean, color = Aplicacion)) +
  geom_errorbar(aes(ymin= mean - sd, ymax = mean + sd), width = .2, position=position_dodge(0.6)) +
  geom_point(position=position_dodge(0.6)) +
  scale_color_manual(name = "Aplicación",
                     values=c(colores[2], colores[1]),
                     ) + # Show dots

  xlab("") +
  ylab("Promedio") +
  coord_flip() +
  labs(fill = "Type") +
  theme_bw()
```



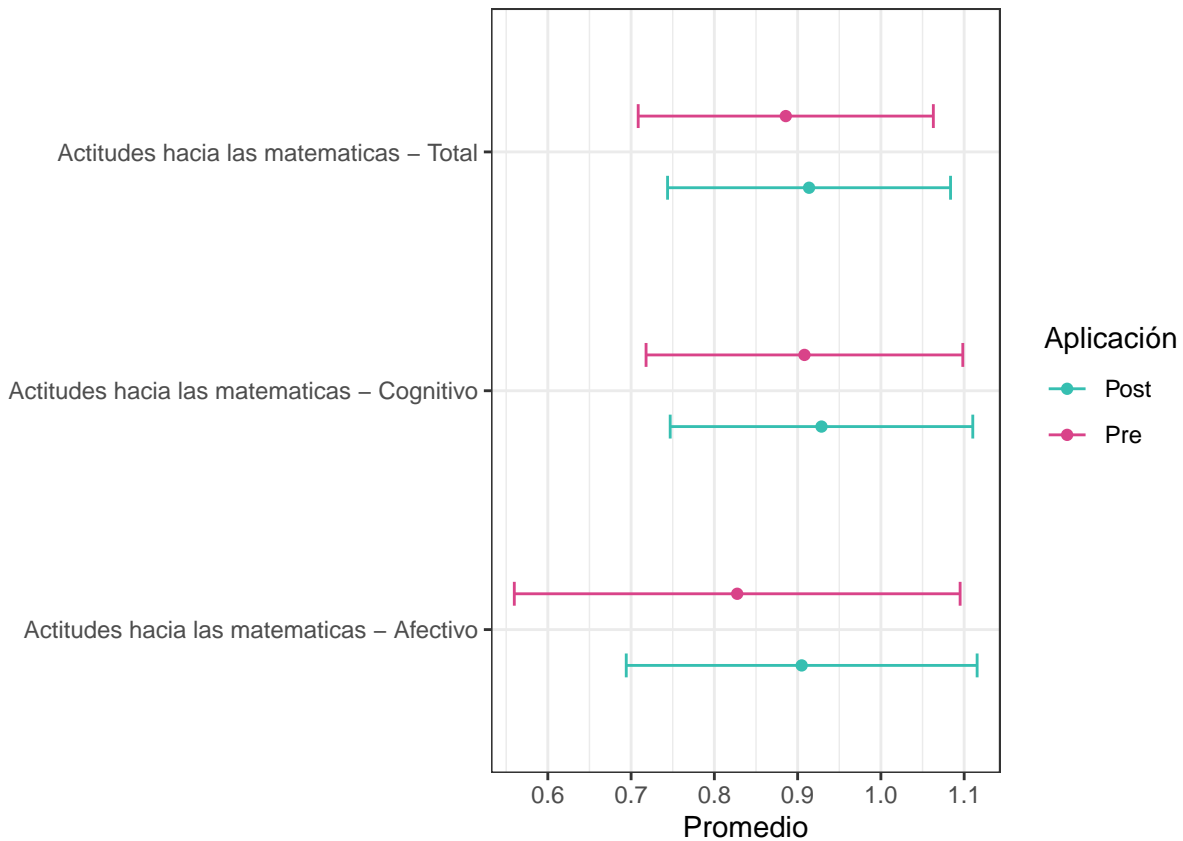
```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/significativas_lenguaje.png")
```

4.4.2 Actitudes hacia las matemáticas

```
significativos %>%
  filter(Area == "Actitudes hacia las matematicas") %>%
  dplyr::select("mean", "sd", "Prueba", "Aplicacion") %>%
  ggplot(aes(x = Prueba, y = mean, color = Aplicacion)) +
  geom_errorbar(aes(ymin= mean - sd, ymax = mean + sd), width = .2, position=position_dodge(0.6)) +
  geom_point(position=position_dodge(0.6)) +
  scale_color_manual(name = "Aplicación",
                     values=c(colores[2], colores[1]),
                     ) + # Show dots

  xlab("") +
  ylab("Promedio") +
  coord_flip() +
  labs(fill = "Type") +
  theme_bw()
```



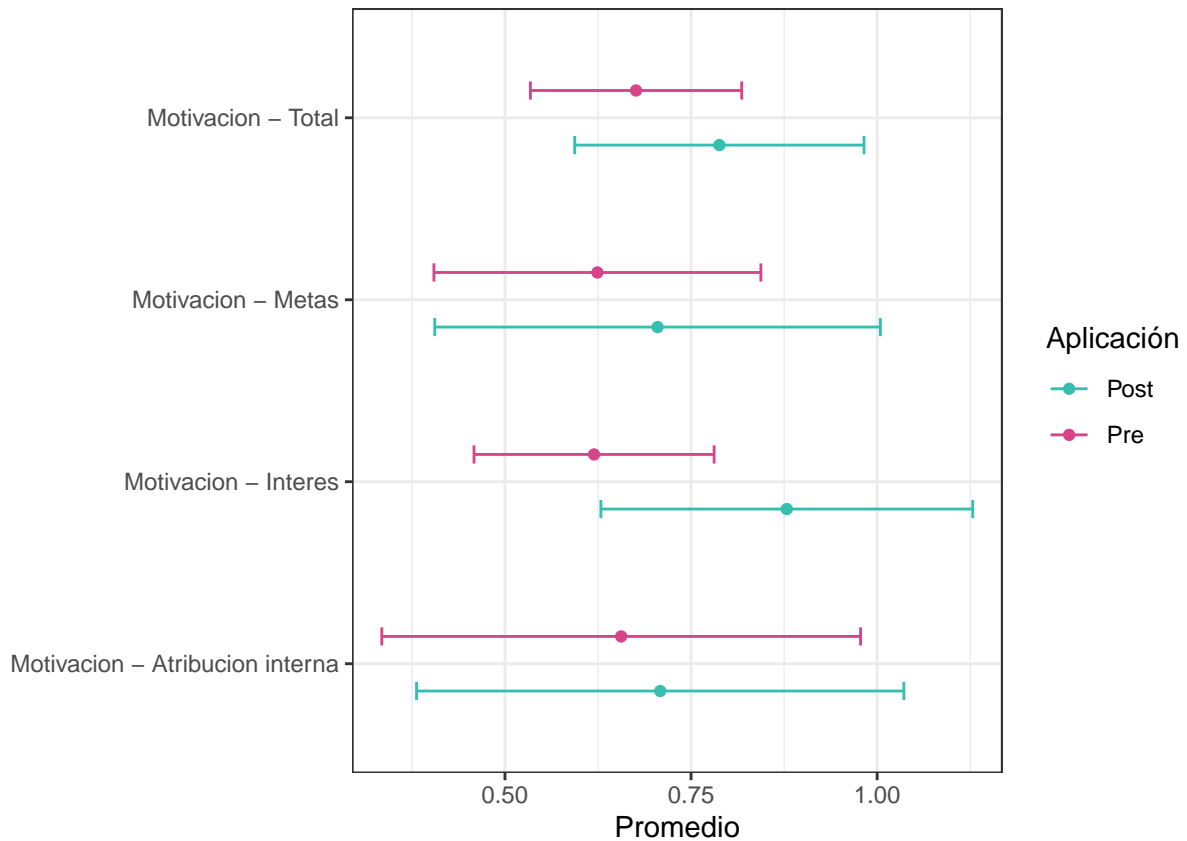
```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/significativas_matematicas.png")
```

4.4.3 Motivación

```
significativos %>%
  filter(Area == "Motivación") %>%
  dplyr::select("mean" , "sd", "Prueba", "Aplicacion") %>%
  ggplot(aes(x = Prueba, y = mean, color = Aplicacion)) +
  geom_errorbar(aes(ymin= mean - sd, ymax = mean + sd), width = .2, position=position_dodge(0.6)) +
  geom_point(position=position_dodge(0.6)) +
  scale_color_manual(name = "Aplicación",
                     values=c(colores[2], colores[1]),
                     ) + # Show dots

  xlab("") +
  ylab("Promedio") +
  coord_flip() +
  labs(fill = "Type") +
  theme_bw()
```



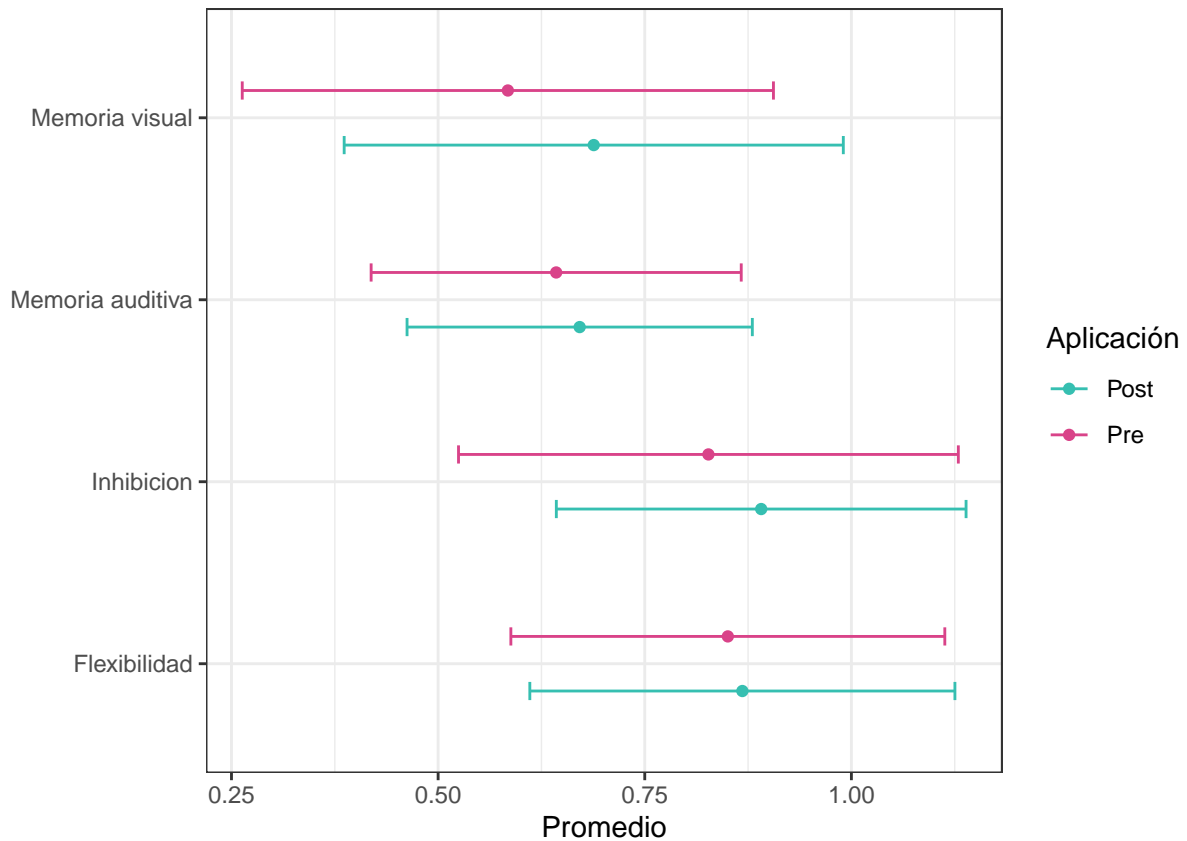
```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/significativas_motivacion.png")
```

4.4.4 Funciones ejecutivas

```
significativos %>%
  filter(Area == "Funciones ejecutivas") %>%
  dplyr::select("mean", "sd", "Prueba", "Aplicacion") %>%
  ggplot(aes(x = Prueba, y = mean, color = Aplicacion)) +
  geom_errorbar(aes(ymin= mean - sd, ymax = mean + sd), width = .2, position=position_dodge(0.6)) +
  geom_point(position=position_dodge(0.6)) +
  scale_color_manual(name = "Aplicación",
                     values=c(colores[2], colores[1]),
                     ) + # Show dots

  xlab("") +
  ylab("Promedio") +
  coord_flip() +
  labs(fill = "Type") +
  theme_bw()
```



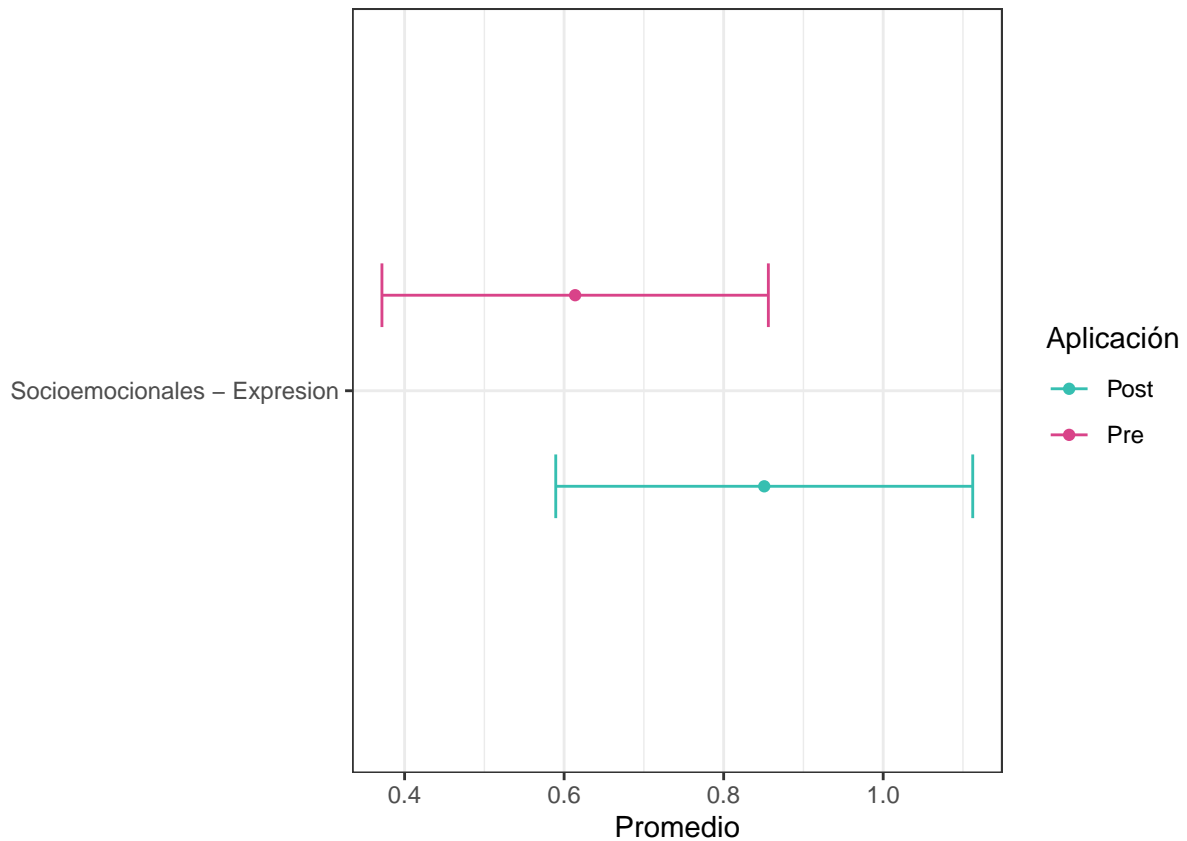
```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/significativas_ejecutivas.png")
```

4.4.5 Habilidades socioemocionales

```
significativos %>%
  filter(Area == "Habilidades socioemocionales") %>%
  dplyr::select("mean", "sd", "Prueba", "Aplicacion") %>%
  ggplot(aes(x = Prueba, y = mean, color = Aplicacion)) +
  geom_errorbar(aes(ymin= mean - sd, ymax = mean + sd), width = .2, position=position_dodge(0.6)) +
  geom_point(position=position_dodge(0.6)) +
  scale_color_manual(name = "Aplicación",
                     values=c(colores[2], colores[1]),
                     ) + # Show dots

  xlab("") +
  ylab("Promedio") +
  coord_flip() +
  labs(fill = "Type") +
  theme_bw()
```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/significativas_socioemocionales.png")
```

4.4.6 Diferencias entre ciudades

```
base = c("Código" , "Ciudad")
```

```
data_ciudades = dplyr::select(actitudes_post, base)
data_ciudades = rbind(data_ciudades, dplyr::select(socioemocional_post, base))
data_ciudades = rbind(data_ciudades, dplyr::select(motivacion_post, base))
data_ciudades = rbind(data_ciudades, dplyr::select(inhibicion_post, base))
data_ciudades = rbind(data_ciudades, dplyr::select(mem_vis_post, base))
data_ciudades = rbind(data_ciudades, dplyr::select(mem_audi_post, base))
```

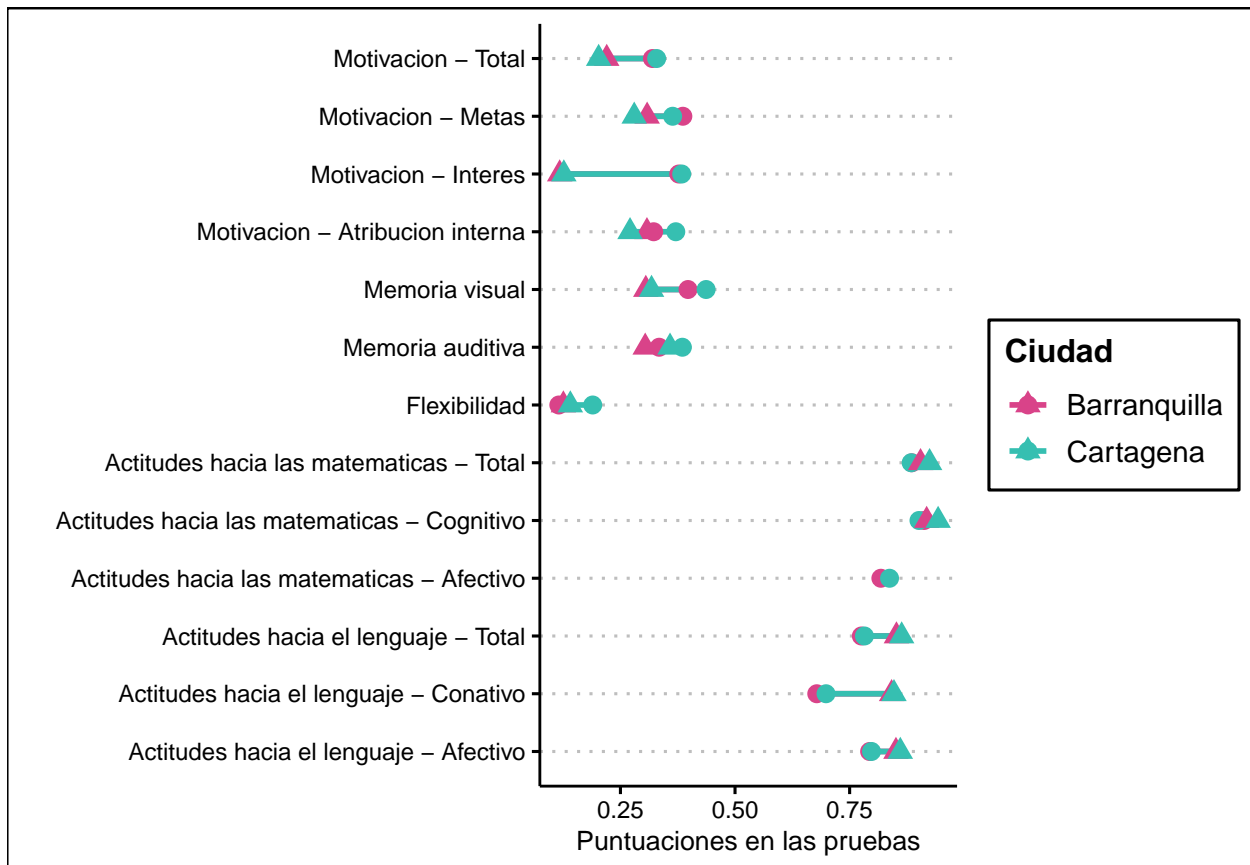
```
pivot_ciudades = dplyr::left_join(pivot_final, data_ciudades, by = "Código")
```

```
pivot_ciudades_viz_2 = pivot_ciudades %>%
  na.omit() %>%
  group_by(Prueba, Ciudad, Tipo) %>%
  summarise(Promedio = mean(score)) %>%
  pivot_wider(names_from = Tipo, values_from = Promedio)
```

```

pivot_ciudades_viz_2 %>%
  filter(Prueba %in% filtro) %>%
  ggplot() +
    geom_segment(aes(x=Prueba, xend=Prueba, y=pre, yend=post, color = Ciudad), size=1) +
    geom_point( aes(x=Prueba, y=pre, color = Ciudad), size=3, shape = 16) +
    geom_point( aes(x=Prueba, y=post, color = Ciudad), size=3, shape = 17 ) +
    coord_flip()+
    theme_clean() +
    scale_color_manual(values=colores[1:2]) +
    xlab("") +
    ylab("Puntuaciones en las pruebas")

```



```
# Guardamos la imagen
```

```
ggsave("../Plots/segunda_aplicacion/significativas_ciudades.png")
```

```
## Warning: Removed 2 rows containing missing values (geom_segment).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```

4.4.7 Diferencias entre ciclos


```

base = c("Código" , "Ciclo")

data_ciclos = dplyr::select(actitudes_post, base)
data_ciclos = rbind(data_ciclos, dplyr::select(socioemocional_post, base))
data_ciclos = rbind(data_ciclos, dplyr::select(motivacion_post, base))
data_ciclos = rbind(data_ciclos, dplyr::select(inhibicion_post, base))
data_ciclos = rbind(data_ciclos, dplyr::select(mem_vis_post, base))
data_ciclos = rbind(data_ciclos, dplyr::select(mem_audi_post, base))

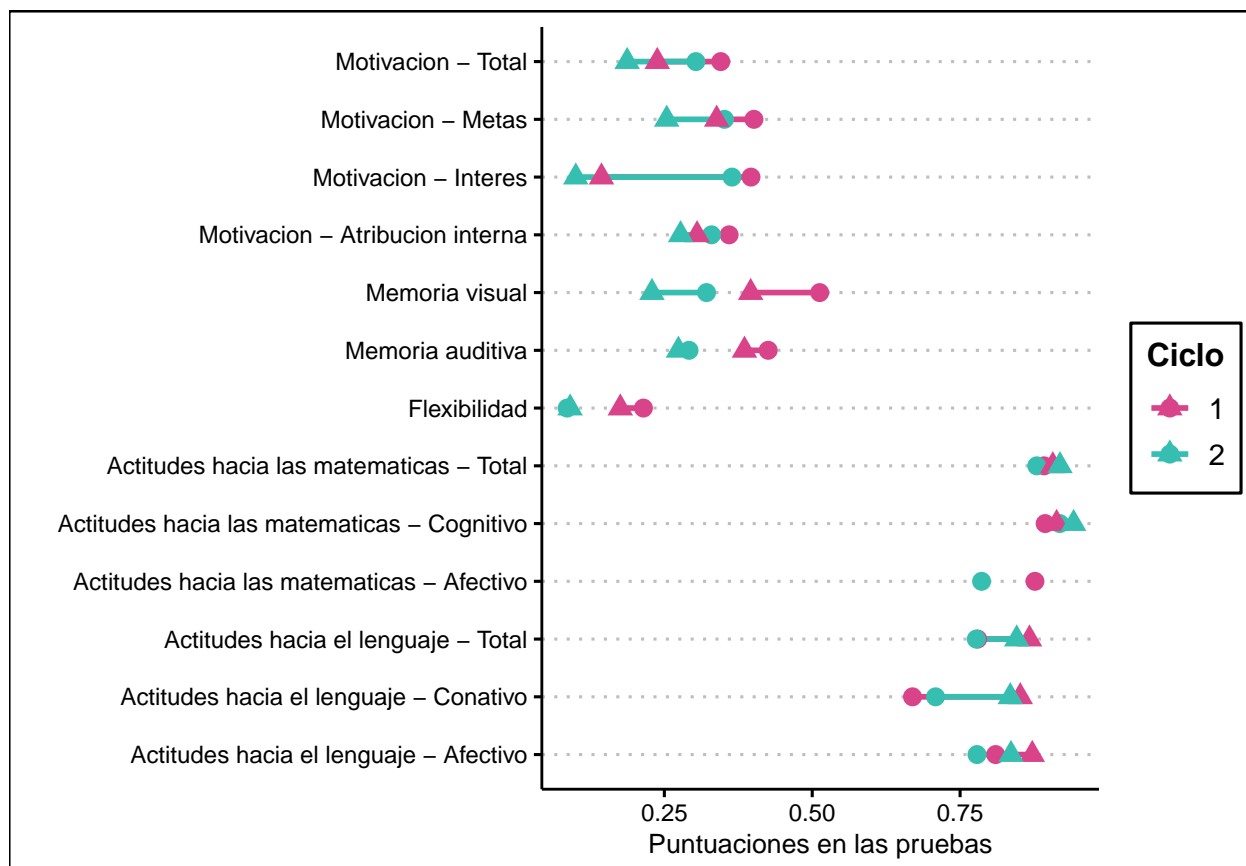
pivot_ciclos = dplyr::left_join(pivot_final, data_ciclos, by = "Código")

pivot_ciclos_viz_2 = pivot_ciclos %>%
  na.omit() %>%
  group_by(Prueba, Ciclo, Tipo) %>%
  summarise(Promedio = mean(score)) %>%
  pivot_wider(names_from = Tipo, values_from = Promedio)

pivot_ciclos_viz_2$Ciclo = as.character(pivot_ciclos_viz_2$Ciclo)

pivot_ciclos_viz_2 %>%
  filter(Prueba %in% filtro) %>%
  ggplot() +
    geom_segment(aes(x=Prueba, xend=Prueba,
                    y=pre, yend=post, color = Ciclo), size=1) +
    geom_point( aes(x=Prueba, y=pre, color = Ciclo), size=3, shape = 16) +
    geom_point( aes(x=Prueba, y=post, color = Ciclo), size=3, shape = 17 ) +
    coord_flip()+
    theme_clean() +
    scale_color_manual(values=colores[1:2]) +
    xlab("") +
    ylab("Puntuaciones en las pruebas")

```



Guardamos la imagen

```
ggsave("../Plots/segunda_aplicacion/significativas_ciclos.png")
```

```
## Warning: Removed 2 rows containing missing values (geom_segment).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```

5 Información de la sesión

```
session_info = devtools::session_info()
dplyr::select(
  tibble::as_tibble(session_info$packages),
  c(package, loadedversion, source)
) %>%
  gt()
```

package	loadedversion	source
admisc	0.27	CRAN (R 4.2.0)
askpass	1.1	CRAN (R 4.2.0)
assertthat	0.2.1	CRAN (R 4.2.0)
backports	1.4.1	CRAN (R 4.2.0)
base64enc	0.1-3	CRAN (R 4.2.0)
boot	1.3-28	CRAN (R 4.1.1)
cachem	1.0.6	CRAN (R 4.1.2)
callr	3.7.0	CRAN (R 4.2.0)
cellranger	1.1.0	CRAN (R 4.2.0)
checkmate	2.1.0	CRAN (R 4.2.0)
cli	3.3.0	CRAN (R 4.1.3)
cluster	2.1.3	CRAN (R 4.1.3)
colorspace	2.0-3	CRAN (R 4.2.0)
crayon	1.5.1	CRAN (R 4.2.0)
curl	4.3.2	CRAN (R 4.1.1)
data.table	1.14.2	CRAN (R 4.2.0)
DBI	1.1.2	CRAN (R 4.2.0)
dcurver	0.9.2	CRAN (R 4.2.0)
deltaPlotR	1.6	CRAN (R 4.2.0)
Deriv	4.1.3	CRAN (R 4.2.0)
desc	1.4.1	CRAN (R 4.2.0)
devtools	2.4.3	CRAN (R 4.1.3)
diffR	5.1	CRAN (R 4.1.3)
digest	0.6.29	CRAN (R 4.1.3)
dplyr	1.0.9	CRAN (R 4.1.3)
DT	0.22	CRAN (R 4.2.0)
effsize	0.8.1	CRAN (R 4.2.0)
ellipsis	0.3.2	CRAN (R 4.1.1)
evaluate	0.15	CRAN (R 4.2.0)
expm	0.999-6	CRAN (R 4.2.0)
extrafont	0.18	CRAN (R 4.2.0)
extrafontdb	1.0	CRAN (R 4.2.0)
fansi	1.0.3	CRAN (R 4.1.3)
farver	2.1.1	CRAN (R 4.1.3)
fastmap	1.1.0	CRAN (R 4.1.1)
foreign	0.8-81	CRAN (R 4.1.1)
Formula	1.2-4	CRAN (R 4.2.0)

fs	1.5.2	CRAN (R 4.1.2)
gargle	1.2.0	CRAN (R 4.2.0)
gdtools	0.2.4	CRAN (R 4.2.0)
generics	0.1.3	CRAN (R 4.1.3)
ggplot2	3.3.6	CRAN (R 4.1.3)
ggtech	0.1.1	Github (ricardo-bion/ggtech@4d6282f230eb1ffe6f44fc9170c572cdeb4b2e11)
ggthemes	4.2.4	CRAN (R 4.1.3)
ggthemr	1.1.0	Github (Mikata-Project/ggthemr@f04aca60b5e0c7a4b6af324a6b3bbcaa9d3)
glue	1.6.2	CRAN (R 4.1.3)
googledrive	2.0.0	CRAN (R 4.2.0)
googlesheets4	1.0.0	CRAN (R 4.2.0)
GPArotation	2022.4-1	CRAN (R 4.2.0)
gridExtra	2.3	CRAN (R 4.2.0)
gt	0.6.0	CRAN (R 4.1.3)
gtable	0.3.1	CRAN (R 4.1.3)
highr	0.9	CRAN (R 4.2.0)
Hmisc	4.7-0	CRAN (R 4.2.0)
hrbrthemes	0.8.6	Github (hrbrmstr/hrbrthemes@3e8d9494a9e0026a3127f6a0df88208511cd07)
htmlTable	2.4.0	CRAN (R 4.2.0)
htmltools	0.5.2	CRAN (R 4.1.1)
htmlwidgets	1.5.4	CRAN (R 4.2.0)
httr	1.4.3	CRAN (R 4.2.0)
jpeg	0.1-9	CRAN (R 4.2.0)
jsonlite	1.8.0	CRAN (R 4.2.0)
knitr	1.39	CRAN (R 4.2.0)
labeling	0.4.2	CRAN (R 4.2.0)
lattice	0.20-44	CRAN (R 4.1.1)
latticeExtra	0.6-29	CRAN (R 4.2.0)
lifecycle	1.0.2	CRAN (R 4.1.3)
likert	1.3.5	CRAN (R 4.2.0)
lme4	1.1-29	CRAN (R 4.2.0)
ltm	1.2-0	CRAN (R 4.1.3)
magrittr	2.0.3	CRAN (R 4.1.3)
MASS	7.3-54	CRAN (R 4.1.1)
Matrix	1.3-4	CRAN (R 4.1.1)
memoise	2.0.1	CRAN (R 4.2.0)
mgcv	1.8-36	CRAN (R 4.1.1)
minqa	1.2.4	CRAN (R 4.2.0)
mirt	1.36.1	CRAN (R 4.2.0)
mnormt	2.0.2	CRAN (R 4.2.0)
msm	1.6.9	CRAN (R 4.2.0)
multilevel	2.7	CRAN (R 4.2.0)
munsell	0.5.0	CRAN (R 4.2.0)
mvtnorm	1.1-3	CRAN (R 4.2.0)
nlme	3.1-152	CRAN (R 4.1.1)
nloptr	2.0.1	CRAN (R 4.2.0)
nnet	7.3-16	CRAN (R 4.1.1)
nortest	1.0-4	CRAN (R 4.2.0)

openssl	2.0.0	CRAN (R 4.2.0)
pbapply	1.5-0	CRAN (R 4.2.0)
permute	0.9-7	CRAN (R 4.2.0)
pillar	1.8.1	CRAN (R 4.1.3)
pkgbuild	1.3.1	CRAN (R 4.2.0)
pkgconfig	2.0.3	CRAN (R 4.2.0)
pkgload	1.2.4	CRAN (R 4.2.0)
plyr	1.8.7	CRAN (R 4.2.0)
png	0.1-7	CRAN (R 4.2.0)
polycor	0.8-1	CRAN (R 4.2.0)
prettyunits	1.1.1	CRAN (R 4.2.0)
processx	3.5.2	CRAN (R 4.1.2)
ps	1.6.0	CRAN (R 4.1.2)
psych	2.2.3	CRAN (R 4.2.0)
psychometric	2.2	CRAN (R 4.1.1)
purrr	0.3.4	CRAN (R 4.1.1)
R6	2.5.1	CRAN (R 4.2.0)
rappdirs	0.3.3	CRAN (R 4.2.0)
RColorBrewer	1.1-3	CRAN (R 4.2.0)
Rcpp	1.0.8.3	CRAN (R 4.2.0)
readxl	1.4.0	CRAN (R 4.1.3)
remotes	2.4.2	CRAN (R 4.2.0)
reshape2	1.4.4	CRAN (R 4.2.0)
rJava	1.0-6	CRAN (R 4.1.2)
rlang	1.0.5	CRAN (R 4.1.3)
rmarkdown	2.14	CRAN (R 4.2.0)
rpart	4.1-15	CRAN (R 4.1.1)
rprojroot	2.0.3	CRAN (R 4.2.0)
rstudioapi	0.14	CRAN (R 4.1.3)
Rttf2pt1	1.3.10	CRAN (R 4.2.0)
scales	1.2.1	CRAN (R 4.1.3)
sessioninfo	1.2.2	CRAN (R 4.2.0)
ShinyItemAnalysis	1.4.1	CRAN (R 4.1.3)
stringi	1.7.6	CRAN (R 4.1.2)
stringr	1.4.1	CRAN (R 4.1.3)
survival	3.2-11	CRAN (R 4.1.1)
systemfonts	1.0.4	CRAN (R 4.2.0)
testthat	3.1.1	CRAN (R 4.1.2)
tibble	3.1.7	CRAN (R 4.1.3)
tidyr	1.2.0	CRAN (R 4.1.3)
tidyselect	1.1.2	CRAN (R 4.2.0)
tmvnsim	1.0-2	CRAN (R 4.2.0)
usethis	2.1.5	CRAN (R 4.2.0)
utf8	1.2.2	CRAN (R 4.1.1)
vctr	0.4.1	CRAN (R 4.1.3)
vegan	2.6-2	CRAN (R 4.2.0)
withr	2.5.0	CRAN (R 4.2.0)
xfun	0.30	CRAN (R 4.1.3)

xlsx	0.6.5	CRAN (R 4.1.3)
xlsxjars	0.6.1	CRAN (R 4.1.1)
xtable	1.8-4	CRAN (R 4.2.0)
yaml	2.3.5	CRAN (R 4.2.0)
