

Análisis de resultados: Primera 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	10
3.1.1.1.1	Total pre	10
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	12
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	13
3.1.1.2	Indicadores psicométricos	14
3.1.1.2.1	Total pre	14
3.1.1.2.2	Total post	15
3.1.1.3	Comparación pre-post	16
3.1.1.3.1	Prueba total	16
3.1.1.3.2	Componente afectivo	19
3.1.1.3.3	Componente Cognitivo	22
3.1.1.3.4	Componente Conativo	25
3.1.2	Actitudes hacia las matemáticas	28
3.1.2.1	Alpha	30
3.1.2.1.1	Total pre	30
3.1.2.1.2	Total post	30
3.1.2.1.3	Componente cognitivo pre	31
3.1.2.1.4	Componente cognitivo post	31
3.1.2.1.5	Componente afectivo pre	32
3.1.2.1.6	Componente afectivo post	32
3.1.2.1.7	Componente conativo pre	32
3.1.2.1.8	Componente conativo post	33
3.1.2.2	Indicadores psicométricos	33
3.1.2.2.1	Total pre	33
3.1.2.2.2	Total post	34
3.1.2.3	Comparación pre-post	35

	3.1.2.3.1	Prueba total	35
	3.1.2.3.2	Componente afectivo	38
	3.1.2.3.3	Componente Cognitivo	41
	3.1.2.3.4	Componente Conativo	44
3.2	Motivación		47
3.2.1	Alpha		50
	3.2.1.1	Total pre	50
	3.2.1.2	Total post	51
	3.2.1.3	Interés pre	51
	3.2.1.4	Interés post	52
	3.2.1.5	Metas pre	52
	3.2.1.6	Metas post	52
	3.2.1.7	Expectativas	53
	3.2.1.8	Atribución interna	53
	3.2.1.9	Expectativa positiva	53
3.2.2	Indicadores psicométricos		53
	3.2.2.1	Total pre	54
	3.2.2.2	Total post	55
3.2.3	Comparación pre-post		55
	3.2.3.1	Prueba total	55
	3.2.3.1.1	Estadísticos de normalidad	56
	3.2.3.1.2	Descriptivos	56
	3.2.3.1.3	Comparación de medias	57
	3.2.3.1.4	Tamaño del efecto	58
	3.2.3.2	Interés	59
	3.2.3.2.1	Estadísticos de normalidad	59
	3.2.3.2.2	Descriptivos	60
	3.2.3.2.3	Comparación de medias	60
	3.2.3.2.4	Tamaño del efecto	61
	3.2.3.3	Metas	62
	3.2.3.3.1	Estadísticos de normalidad	62
	3.2.3.3.2	Descriptivos	63
	3.2.3.3.3	Comparación de medias	63
	3.2.3.3.4	Tamaño del efecto	64
	3.2.3.4	Atribución externa	65
	3.2.3.4.1	Estadísticos de normalidad	65
	3.2.3.4.2	Descriptivos	66
	3.2.3.4.3	Comparación de medias	66
	3.2.3.4.4	Tamaño del efecto	67
	3.2.3.5	Expectativas positivas	68
	3.2.3.5.1	Estadísticos de normalidad	68
	3.2.3.5.2	Descriptivos	69
	3.2.3.5.3	Comparación de medias	69
	3.2.3.5.4	Tamaño del efecto	71
3.3	Funciones ejecutivas		71
3.3.1	Memoria auditiva		71
	3.3.1.1	Alpha	74
	3.3.1.1.1	Pre	74

	3.3.1.1.2	Post	75
	3.3.1.2	Indicadores psicométricos	76
	3.3.1.2.1	Pre	76
	3.3.1.2.2	Post	78
	3.3.1.3	Comparación pre-post	80
	3.3.1.3.1	Estadísticos de normalidad	80
	3.3.1.3.2	Descriptivos	81
	3.3.1.3.3	Comparación de medias	81
	3.3.1.3.4	Tamaño del efecto	83
3.3.2		Memoria visual	83
	3.3.2.1	Alpha	85
	3.3.2.1.1	Pre	85
	3.3.2.1.2	Post	86
	3.3.2.2	Indicadores psicométricos	87
	3.3.2.2.1	Pre	87
	3.3.2.2.2	Post	89
	3.3.2.3	Comparación pre-post	91
	3.3.2.3.1	Estadísticos de normalidad	91
	3.3.2.3.2	Descriptivos	92
	3.3.2.3.3	Comparación de medias	92
	3.3.2.3.4	Tamaño del efecto	94
3.3.3		Inhibición	94
	3.3.3.1	Alpha	96
	3.3.3.1.1	Pre	96
	3.3.3.1.2	Post	96
	3.3.3.2	Indicadores psicométricos	97
	3.3.3.2.1	Pre	98
	3.3.3.2.2	Post	99
	3.3.3.3	Comparación pre-post	100
	3.3.3.3.1	Estadísticos de normalidad	100
	3.3.3.3.2	Descriptivos	100
	3.3.3.3.3	Comparación de medias	101
	3.3.3.3.4	Tamaño del efecto	102
3.3.4		Flexibilidad	103
	3.3.4.1	Alpha	104
	3.3.4.1.1	Pre	104
	3.3.4.1.2	Post	105
	3.3.4.2	Indicadores psicométricos	106
	3.3.4.2.1	Pre	106
	3.3.4.2.2	Post	107
	3.3.4.3	Comparación pre-post	108
	3.3.4.3.1	Estadísticos de normalidad	108
	3.3.4.3.2	Descriptivos	109
	3.3.4.3.3	Comparación de medias	109
	3.3.4.3.4	Tamaño del efecto	111
3.4		Socioemocionales	111
	3.4.1	Alpha	115
	3.4.1.1	Total pre	116

3.4.1.2	Total post	116
3.4.1.3	Regulación pre	117
3.4.1.4	Regulación post	117
3.4.1.5	Expresión pre	118
3.4.1.6	Expresión post	118
3.4.1.7	Reconocimiento pre	118
3.4.1.8	Reconocimiento post	119
3.4.2	Indicadores psicométricos	119
3.4.2.0.1	Total pre	120
3.4.2.0.2	Total post	121
3.4.3	Comparación pre-post	121
3.4.3.1	Regulación	122
3.4.3.1.1	Estadísticos de normalidad	122
3.4.3.1.2	Descriptivos	123
3.4.3.1.3	Comparación de medias	123
3.4.3.1.4	Tamaño del efecto	124
3.4.3.2	Reconocimiento	126
3.4.3.2.1	Estadísticos de normalidad	126
3.4.3.2.2	Descriptivos	126
3.4.3.2.3	Comparación de medias	127
3.4.3.2.4	Tamaño del efecto	128
3.4.3.3	Expresión	130
3.4.3.3.1	Estadísticos de normalidad	130
3.4.3.3.2	Descriptivos	130
3.4.3.3.3	Comparación de medias	131
3.4.3.3.4	Tamaño del efecto	132
4	Resúmenes de datos	134
4.1	Alpha de las pruebas	134
4.2	Descriptivos	135
4.3	Indicadores psicométricos	136
4.4	Pruebas pre-post	141
4.4.1	Actitudes hacia el lenguaje	143
4.4.2	Actitudes hacia las matemáticas	144
4.4.3	Motivación	145
4.4.4	Funciones ejecutivas	146
4.4.5	Habilidades socioemocionales	147
4.4.6	Diferencias entre ciudades	148
4.4.7	Diferencias entre ciclos	150
5	Información de la sesión	152

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-Primera_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",  
               "/1Ry73ckzruTQxtDnkCSscs16M3cv3u9XgJlcg4sN94BE/edit?usp=sharing",  
               sep = "")
```

```
# post
```

```
url_post = paste("https://docs.google.com/spreadsheets/d/",  
                "1tFkxbH5N9HMr5qRA-VF4JXh-MAJqW38-iidB4QBuKU4/edit#gid=1877736074",  
                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[,1:15]

col_items = paste(rep("Grupo", 9), rep(1:3, 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] 1161 15

actitudes_pre[,7:15] = apply(actitudes_pre[,7:15], 2, function(x) str_to_upper(x))

actitudes_pre[,7:15] = apply(actitudes_pre[,7:15], 2,
                             function(x) {ifelse(x == "B", 1, 0)})
```

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[,1:15]

col_items = paste(rep("Grupo", 9), rep(1:3, each = 3), rep(paste("_", rep(1:3, 3))))

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

actitudes_post[,7:15] = apply(actitudes_post[,7:15], 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[,7:15] = apply(actitudes_post[,7:15], 2, function(x) str_to_upper(x))
```



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

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

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 post
actitudes_lenguaje_post[,7:15] = calificacion(actitudes_lenguaje_post[,7:15],
                                             claves_lenguaje)
```

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

```
# Calificación

# Total

actitudes_lenguaje_pre$Total_pre = apply(actitudes_lenguaje_pre[,7:15], 1,
                                          function(x) sum(x, na.rm= FALSE))
actitudes_lenguaje_post$Total_post = apply(actitudes_lenguaje_post[,7:15], 1,
                                             function(x) sum(x, na.rm= FALSE))

# Afectivo

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

# Cognitivo

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

# Conativo

actitudes_lenguaje_pre$Conativo_pre = apply(actitudes_lenguaje_pre[,c(9, 12, 15)], 1,
                                             function(x) sum(x, na.rm= FALSE))
actitudes_lenguaje_post$Conativo_post = apply(actitudes_lenguaje_post[,c(9, 12, 15)], 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[,7:15])
```

3.1.1.1.1 Total pre

```
## [1] 0.80124
```

```
for(i in seq(length(colnames(actitudes_lenguaje_pre[,7:15])))){
  x = alpha(actitudes_lenguaje_pre[,7:15][,-i])
  print(paste("El índice de confiabilidad cambia a", round(x, 2),
              "al eliminar el ítem",
              colnames(actitudes_lenguaje_pre[,7:15])[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.8 al eliminar el ítem Grupo 1 _ 1"
## [1] "El índice de confiabilidad cambia a 0.79 al eliminar el ítem Grupo 1 _ 2"
## [1] "El índice de confiabilidad cambia a 0.78 al eliminar el ítem Grupo 1 _ 3"
## [1] "El índice de confiabilidad cambia a 0.78 al eliminar el ítem Grupo 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.79 al eliminar el ítem Grupo 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.78 al eliminar el ítem Grupo 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.78 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 0.77 al eliminar el ítem Grupo 3 _ 3"
```

```
alpha(actitudes_lenguaje_post[,7:15])
```

3.1.1.1.2 Total post

```
## [1] 0.79536
```

```
for(i in seq(length(colnames(actitudes_lenguaje_post[,7:15])))){
  x = alpha(actitudes_lenguaje_post[,7:15][,-i])
  print(paste("El índice de confiabilidad cambia a", round(x, 2), "al eliminar el ítem",
              colnames(actitudes_lenguaje_post[,7:15])[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.79 al eliminar el ítem Grupo 1 _ 1"
## [1] "El índice de confiabilidad cambia a 0.78 al eliminar el ítem Grupo 1 _ 2"
## [1] "El índice de confiabilidad cambia a 0.79 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 0.77 al eliminar el ítem Grupo 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.77 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 0.77 al eliminar el ítem Grupo 3 _ 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[,7:15])))

alfa = rbind(alfa,
             c("Actitudes lenguaje post",
               alpha(actitudes_lenguaje_post[,7:15])))

```

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

3.1.1.1.3 Componente cognitivo pre

```
## [1] 0.53779
```

```

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

```

```
## [1] "El índice de confiabilidad cambia a 0.469734544540396 al eliminar el ítem Grupo 1 _ 2"
```

```
## [1] "El índice de confiabilidad cambia a 0.422874341610233 al eliminar el ítem Grupo 2 _ 2"
```

```
## [1] "El índice de confiabilidad cambia a 0.408609400059248 al eliminar el ítem Grupo 3 _ 2"
```

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

3.1.1.1.4 Componente cognitivo post

```
## [1] 0.46791
```

```

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

```

```
## [1] "El índice de confiabilidad cambia a 0.386946386946387 al eliminar el ítem Grupo 1 _ 2"
```

```
## [1] "El índice de confiabilidad cambia a 0.455425846279879 al eliminar el ítem Grupo 2 _ 2"
```

```
## [1] "El índice de confiabilidad cambia a 0.288 al eliminar el ítem Grupo 3 _ 2"
```

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

3.1.1.1.5 Componente afectivo pre

```
## [1] 0.4658
```

```

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

```

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

```
## [1] "El índice de confiabilidad cambia a 0.420100382409178 al eliminar el ítem Grupo 1 _ 1"
## [1] "El índice de confiabilidad cambia a 0.278155418026868 al eliminar el ítem Grupo 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.407700609405394 al eliminar el ítem Grupo 3 _ 1"
```

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

3.1.1.1.6 Componente afectivo post

```
## [1] 0.51561
```

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

```
## [1] "El índice de confiabilidad cambia a 0.505764346920505 al eliminar el ítem Grupo 1 _ 1"
## [1] "El índice de confiabilidad cambia a 0.360369881109643 al eliminar el ítem Grupo 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.335934134316213 al eliminar el ítem Grupo 3 _ 1"
```

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

3.1.1.1.7 Componente conativo pre

```
## [1] 0.56206
```

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

```
## [1] "El índice de confiabilidad cambia a 0.443895859313364 al eliminar el ítem Grupo 1 _ 3"
## [1] "El índice de confiabilidad cambia a 0.421728685591018 al eliminar el ítem Grupo 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.524196670538134 al eliminar el ítem Grupo 3 _ 3"
```

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

3.1.1.1.8 Componente conativo post

```
## [1] 0.50883
```

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

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

```
## [1] "El índice de confiabilidad cambia a 0.463281528057895 al eliminar el ítem Grupo 1 _ 3"
## [1] "El índice de confiabilidad cambia a 0.437565184412629 al eliminar el ítem Grupo 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.328165530554382 al eliminar el ítem Grupo 3 _ 3"
```

3.1.1.2 Indicadores psicométricos

```
analitem = item.exam(actitudes_lenguaje_pre[,7:15], 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
0.444	0.535	0.377	0.730	0.538	0.238
0.347	0.559	0.443	0.860	0.369	0.194
0.406	0.645	0.524	0.792	0.518	0.262
0.403	0.599	0.469	0.797	0.456	0.241
0.327	0.569	0.463	0.879	0.303	0.186
0.382	0.619	0.501	0.823	0.446	0.237
0.473	0.670	0.533	0.664	0.723	0.317
0.471	0.712	0.588	0.669	0.764	0.335

0.459	0.683	0.553	0.700	0.703	0.313
-------	-------	-------	-------	-------	-------

```
# 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[,7:15], 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
0.275	0.506	0.399	0.918	0.224	0.139
0.335	0.552	0.426	0.871	0.333	0.185
0.380	0.503	0.349	0.825	0.388	0.191
0.454	0.690	0.548	0.710	0.727	0.313

0.455	0.683	0.538	0.708	0.733	0.311
0.439	0.682	0.543	0.740	0.655	0.299
0.354	0.648	0.532	0.853	0.400	0.229
0.319	0.633	0.528	0.885	0.309	0.202
0.350	0.645	0.531	0.857	0.370	0.226

```
# 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 = 0.209, 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 = 0.25, 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	434	0.77163	0.26358	0.88889	0.81418	0.16473	0	1	1
Total_post	434	0.82309	0.23157	0.88889	0.86654	0.16473	0	1	1

Comparación de medias

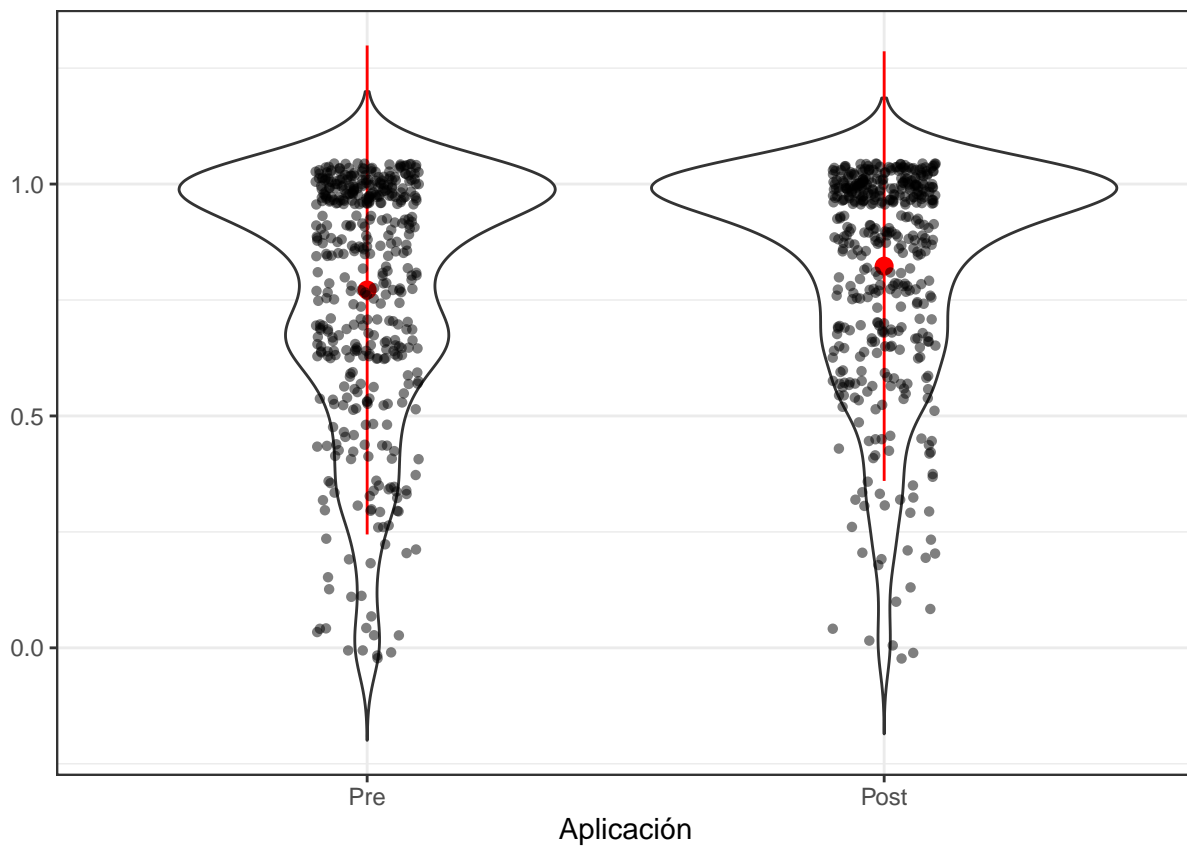
```
comparacion =  
  wilcox.test(  
    x = pre_post$Total_pre,  
    y = pre_post$Total_post,  
    alternative = "two.sided",  
    mu = 0,  
    var.equal = TRUE,  
    paired = TRUE,  
    conf.level = 0.95  
  )
```

comparacion

```
##  
## Wilcoxon signed rank test with continuity correction  
##  
## data: pre_post$Total_pre and pre_post$Total_post  
## V = 16511, p-value = 0.0001  
## 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/primer_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: 0.20706 (small)
## 95 percent confidence interval:
##   lower   upper
## 0.093858 0.320267
```

```
# 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)

```

```

# Adicionalmente, 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 = 0.297, 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 = 0.391, 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	434	0.73733	0.31000	0.66667	0.78161	0.4942	0	1	1
Afectivo_post	434	0.83180	0.26248	1.00000	0.88410	0.0000	0	1	1

Comparación de medias

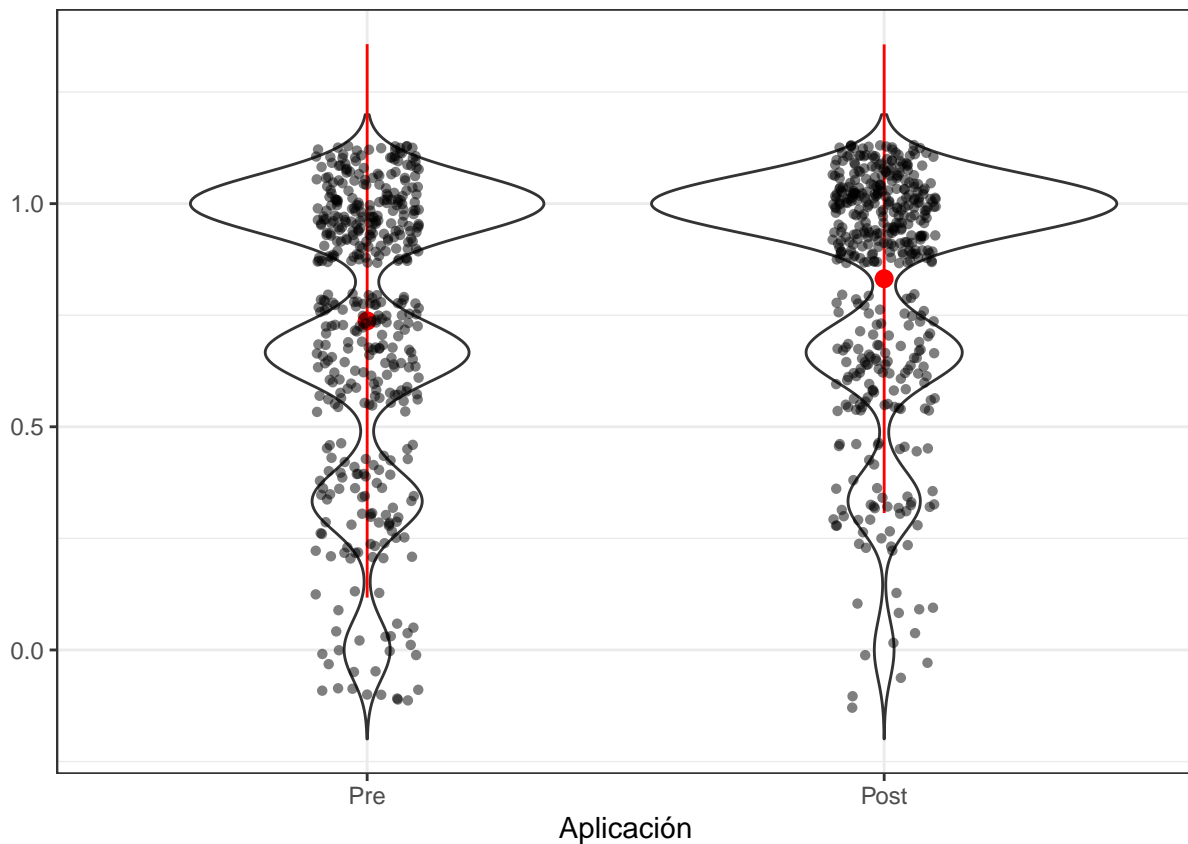
```
comparacion =
  wilcox.test(
    x      = pre_post$Afectivo_pre,
    y      = pre_post$Afectivo_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )

comparacion
```

```
##
## Wilcoxon signed rank test with continuity correction
##
## data:  pre_post$Afectivo_pre and pre_post$Afectivo_post
## V = 7034, p-value = 0.000000024
## 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/primer_aaplicacion/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: 0.32817 (small)
## 95 percent confidence interval:
```

```
##      lower      upper
## 0.20951 0.44683

# 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 = 0.366, 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 = 0.375, 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	434	0.80722	0.27923	1	0.85824	0	0	1	1
Cognitivo_post	434	0.82565	0.25744	1	0.87548	0	0	1	1

Comparación de medias

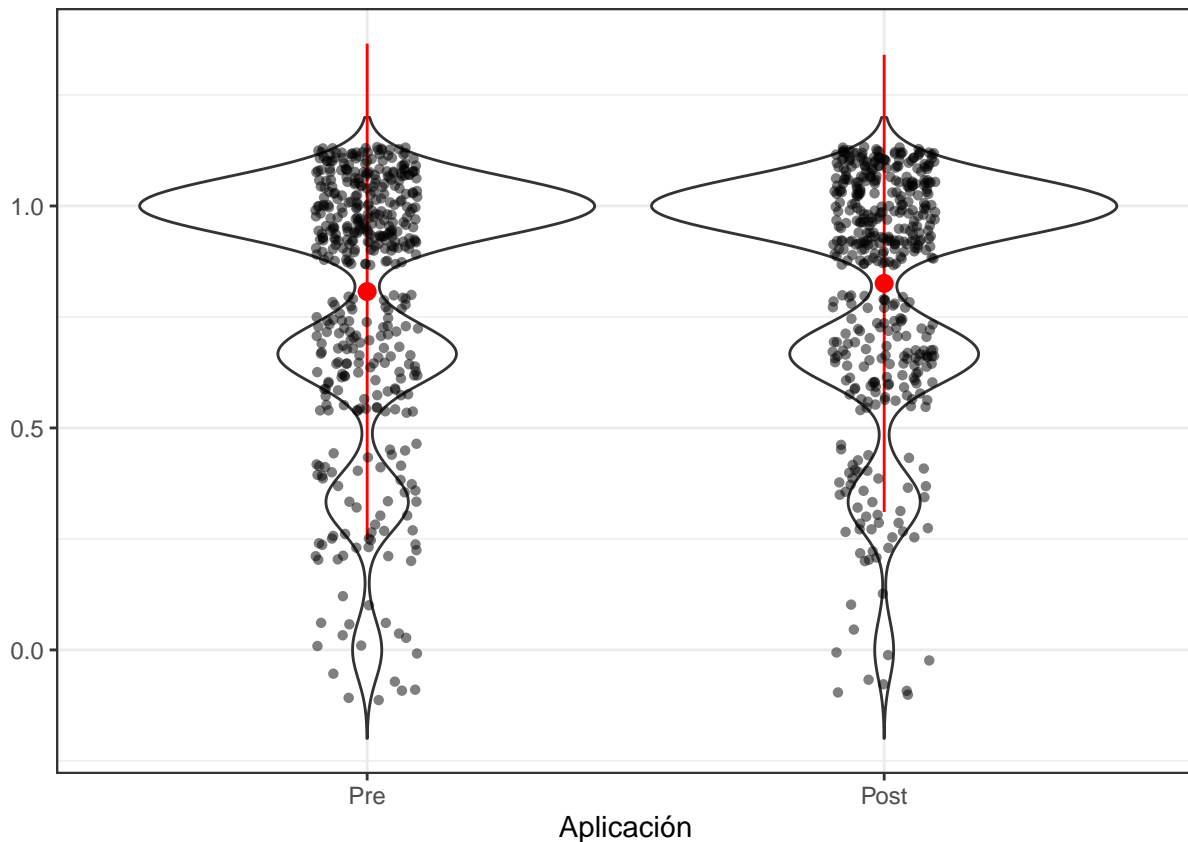
```
comparacion =  
  wilcox.test(  
    x      = pre_post$Cognitivo_pre,  
    y      = pre_post$Cognitivo_post,  
    alternative = "two.sided",  
    mu      = 0,  
    var.equal = TRUE,  
    paired   = TRUE,  
    conf.level = 0.95  
  )
```

comparacion

```
##  
## Wilcoxon signed rank test with continuity correction  
##  
## data: pre_post$Cognitivo_pre and pre_post$Cognitivo_post  
## V = 9908, p-value = 0.37  
## 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/primer_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: 0.068602 (negligible)
## 95 percent confidence interval:
##    lower    upper
## -0.047008  0.184211
```

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 = 0.332, 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 = 0.364, 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	434	0.77035	0.30436	1	0.82184	0	0	1	1
Conativo_post	434	0.81183	0.27499	1	0.86494	0	0	1	1

Comparación de medias

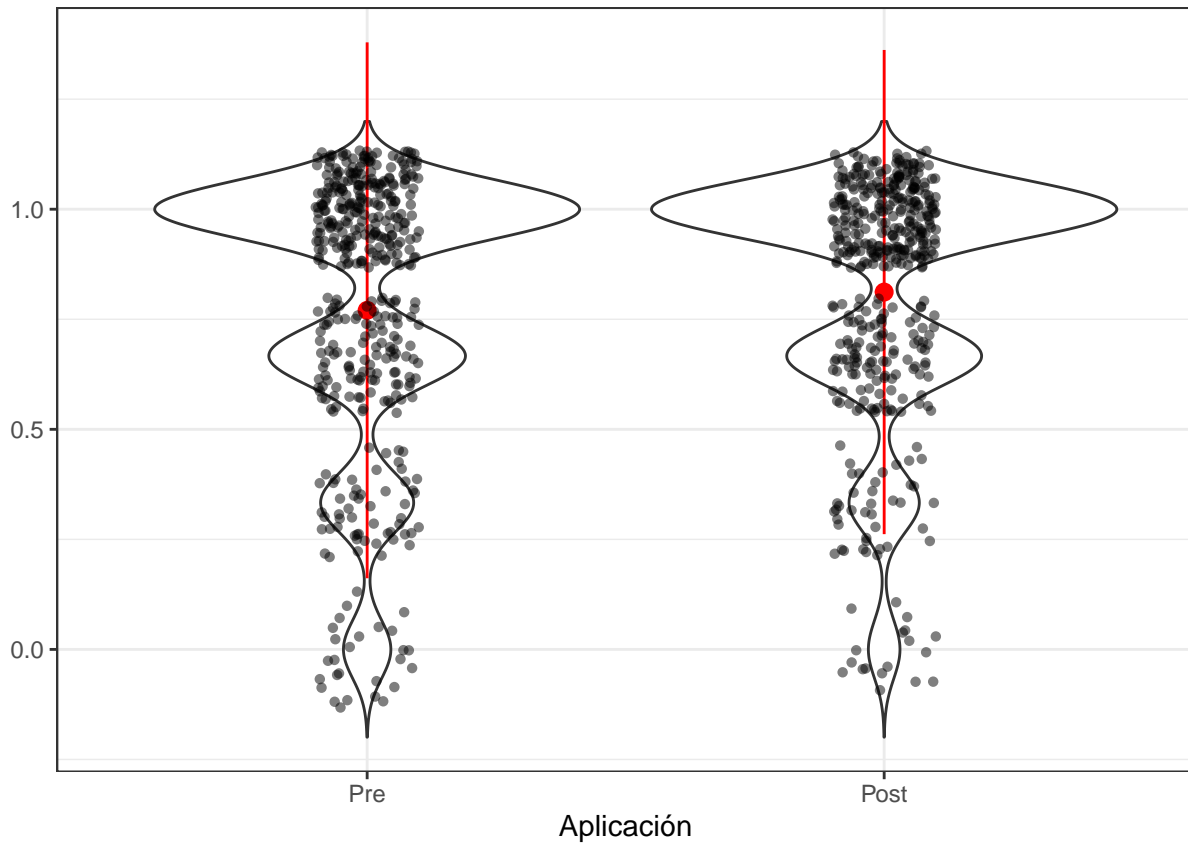
```
comparacion =
  wilcox.test(
    x      = pre_post$Conativo_pre,
    y      = pre_post$Conativo_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )
```

```
comparacion
```

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Conativo_pre and pre_post$Conativo_post
## V = 9699, p-value = 0.015
## 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/primer_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: 0.14286 (negligible)  
## 95 percent confidence interval:  
##   lower   upper  
## 0.027998 0.257722
```

```
# 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")

```

```

actitudes_matematicas_post[,7:15] =
  calificacion(actitudes_matematicas_post[,7:15],
    claves_matematicas)

```

Calificación

Total

```

actitudes_matematicas_pre$Total_pre =
  apply(actitudes_matematicas_pre[,7:15], 1,
    function(x) sum(x, na.rm= FALSE))

actitudes_matematicas_post$Total_post =
  apply(actitudes_matematicas_post[,7:15], 1,
    function(x) sum(x, na.rm= FALSE))

```

Afectivo

```

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

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

```

```

# Cognitivo

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

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

# Conativo

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

actitudes_matematicas_post$Conativo_post =
  apply(actitudes_matematicas_post[,c(9, 12, 15)], 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")

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 las matematicas - Total",
                                           "Afectivo" = "Actitudes hacia las matematicas - Afect.",
                                           "Cognitivo" = "Actitudes hacia las matematicas - Cogn.",
                                           "Conativo" = "Actitudes hacia las matematicas - Conat."))

pivot_final = rbind(pivot_final, pivot_to_bind)

```

3.1.2.1 Alpha

```
alpha(actitudes_matematicas_pre[,7:15])
```

3.1.2.1.1 Total pre

```
## [1] 0.86738
```

```

for(i in seq(length(colnames(actitudes_matematicas_pre[,7:15])))){
  x = alpha(actitudes_matematicas_pre[,7:15][,-i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(actitudes_matematicas_pre[,7:15])[i]))
}

```

```

## [1] "El índice de confiabilidad cambia a 0.866927870344236 al eliminar el ítem Grupo 1 _ 1"
## [1] "El índice de confiabilidad cambia a 0.855002453978297 al eliminar el ítem Grupo 1 _ 2"
## [1] "El índice de confiabilidad cambia a 0.859822098694714 al eliminar el ítem Grupo 1 _ 3"
## [1] "El índice de confiabilidad cambia a 0.850284551708851 al eliminar el ítem Grupo 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.850608545831597 al eliminar el ítem Grupo 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.847485051486142 al eliminar el ítem Grupo 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.854742110885768 al eliminar el ítem Grupo 3 _ 1"
## [1] "El índice de confiabilidad cambia a 0.850128758539855 al eliminar el ítem Grupo 3 _ 2"
## [1] "El índice de confiabilidad cambia a 0.84367079172633 al eliminar el ítem Grupo 3 _ 3"

```

```
alpha(actitudes_matematicas_post[,7:15])
```

3.1.2.1.2 Total post

```
## [1] 0.83482
```

```

for(i in seq(length(colnames(actitudes_matematicas_post[,7:15])))){
  x = alpha(actitudes_matematicas_post[,7:15][,-i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(actitudes_matematicas_post[,7:15])[i]))
}

```

```
## [1] "El índice de confiabilidad cambia a 0.832877002611357 al eliminar el ítem Grupo 1 _ 1"
## [1] "El índice de confiabilidad cambia a 0.822848131203732 al eliminar el ítem Grupo 1 _ 2"
## [1] "El índice de confiabilidad cambia a 0.827904824319074 al eliminar el ítem Grupo 1 _ 3"
## [1] "El índice de confiabilidad cambia a 0.811013698126052 al eliminar el ítem Grupo 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.813988819360638 al eliminar el ítem Grupo 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.810171881842676 al eliminar el ítem Grupo 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.817386151569538 al eliminar el ítem Grupo 3 _ 1"
## [1] "El índice de confiabilidad cambia a 0.812197977425821 al eliminar el ítem Grupo 3 _ 2"
## [1] "El índice de confiabilidad cambia a 0.812435498348564 al eliminar el ítem Grupo 3 _ 3"
```

Amacemos los valores

```
alfa = rbind(alfa,
             c("Actitudes matemáticas pre",
               alpha(actitudes_matematicas_pre[,7:15])))

alfa = rbind(alfa,
             c("Actitudes matemáticas post",
               alpha(actitudes_matematicas_post[,7:15])))
```

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

3.1.2.1.3 Componente cognitivo pre

```
## [1] 0.68445
```

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

```
## [1] "El índice de confiabilidad cambia a 0.661498095641134 al eliminar el ítem Grupo 1 _ 2"
## [1] "El índice de confiabilidad cambia a 0.51016325986322 al eliminar el ítem Grupo 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.593103719430683 al eliminar el ítem Grupo 3 _ 2"
```

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

3.1.2.1.4 Componente cognitivo post

```
## [1] 0.64522
```

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

```
## [1] "El índice de confiabilidad cambia a 0.594075988132835 al eliminar el ítem Grupo 1 _ 2"
## [1] "El índice de confiabilidad cambia a 0.472544763971785 al eliminar el ítem Grupo 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.572413793103448 al eliminar el ítem Grupo 3 _ 2"
```

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

3.1.2.1.5 Componente afectivo pre

```
## [1] 0.62622
```

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

```
## [1] "El índice de confiabilidad cambia a 0.586827548172536 al eliminar el ítem Grupo 1 _ 1"
## [1] "El índice de confiabilidad cambia a 0.493480662983425 al eliminar el ítem Grupo 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.509852012660955 al eliminar el ítem Grupo 3 _ 1"
```

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

3.1.2.1.6 Componente afectivo post

```
## [1] 0.53699
```

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

```
## [1] "El índice de confiabilidad cambia a 0.483615540914968 al eliminar el ítem Grupo 1 _ 1"
## [1] "El índice de confiabilidad cambia a 0.365691621219633 al eliminar el ítem Grupo 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.462860546243512 al eliminar el ítem Grupo 3 _ 1"
```

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

3.1.2.1.7 Componente conativo pre

```
## [1] 0.70346
```

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

```
## [1] "El índice de confiabilidad cambia a 0.720924499229584 al eliminar el ítem Grupo 1 _ 3"
```



```
## [1] "El índice de confiabilidad cambia a 0.570734262681528 al eliminar el ítem Grupo 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.544375 al eliminar el ítem Grupo 3 _ 3"
```

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

3.1.2.1.8 Componente conativo post

```
## [1] 0.59954
```

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

```
## [1] "El índice de confiabilidad cambia a 0.616940335695273 al eliminar el ítem Grupo 1 _ 3"
## [1] "El índice de confiabilidad cambia a 0.399916989147552 al eliminar el ítem Grupo 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.48302367461677 al eliminar el ítem Grupo 3 _ 3"
```

3.1.2.2 Indicadores psicométricos

```
analitem = item.exam(actitudes_matematicas_pre[,7:15], 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
0.416	0.611	0.476	0.777	0.539	0.254

0.327	0.673	0.582	0.878	0.366	0.220
0.396	0.655	0.537	0.805	0.524	0.259
0.315	0.721	0.643	0.889	0.319	0.227
0.331	0.716	0.633	0.875	0.356	0.237
0.333	0.746	0.669	0.873	0.361	0.248
0.396	0.696	0.587	0.805	0.513	0.276
0.360	0.723	0.633	0.847	0.440	0.260
0.375	0.777	0.697	0.831	0.497	0.291

```
# 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[,7:15], 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
0.398	0.586	0.433	0.803	0.512	0.233

0.298	0.604	0.497	0.902	0.256	0.180
0.381	0.606	0.465	0.824	0.477	0.231
0.334	0.705	0.606	0.873	0.366	0.235
0.313	0.681	0.584	0.890	0.314	0.213
0.336	0.711	0.613	0.871	0.360	0.239
0.344	0.661	0.547	0.863	0.390	0.227
0.348	0.699	0.593	0.859	0.390	0.243
0.316	0.693	0.598	0.888	0.337	0.219

```
# 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)
```

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 = 0.287, 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 = 0.312, 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	468	0.84639	0.25118	1	0.90632	0	0	1	1
Total_post	468	0.87132	0.22117	1	0.92494	0	0	1	1

Comparación de medias

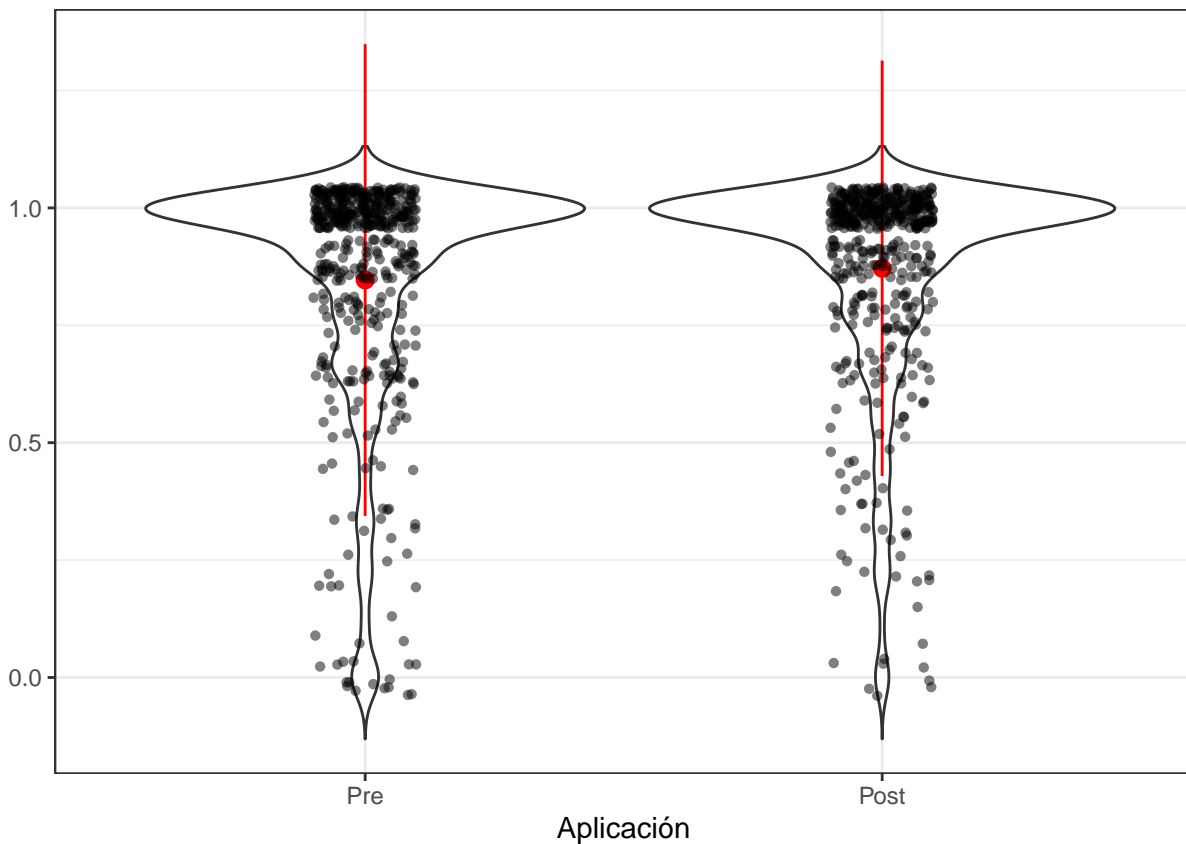
```
comparacion =
  wilcox.test(
    x      = pre_post$Total_pre,
    y      = pre_post$Total_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )

comparacion
```

```
##
## Wilcoxon signed rank test with continuity correction
##
## data:  pre_post$Total_pre and pre_post$Total_post
## V = 12310, p-value = 0.016
## 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/primer_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: 0.10503 (negligible)
## 95 percent confidence interval:
```

```
##      lower      upper
## 0.0071053 0.2029640

# 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 = 0.391, 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 = 0.415, 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	468	0.82550	0.28723	1	0.88918	0	0	1	1
Afectivo_post	468	0.85399	0.25495	1	0.91312	0	0	1	1

Comparación de medias

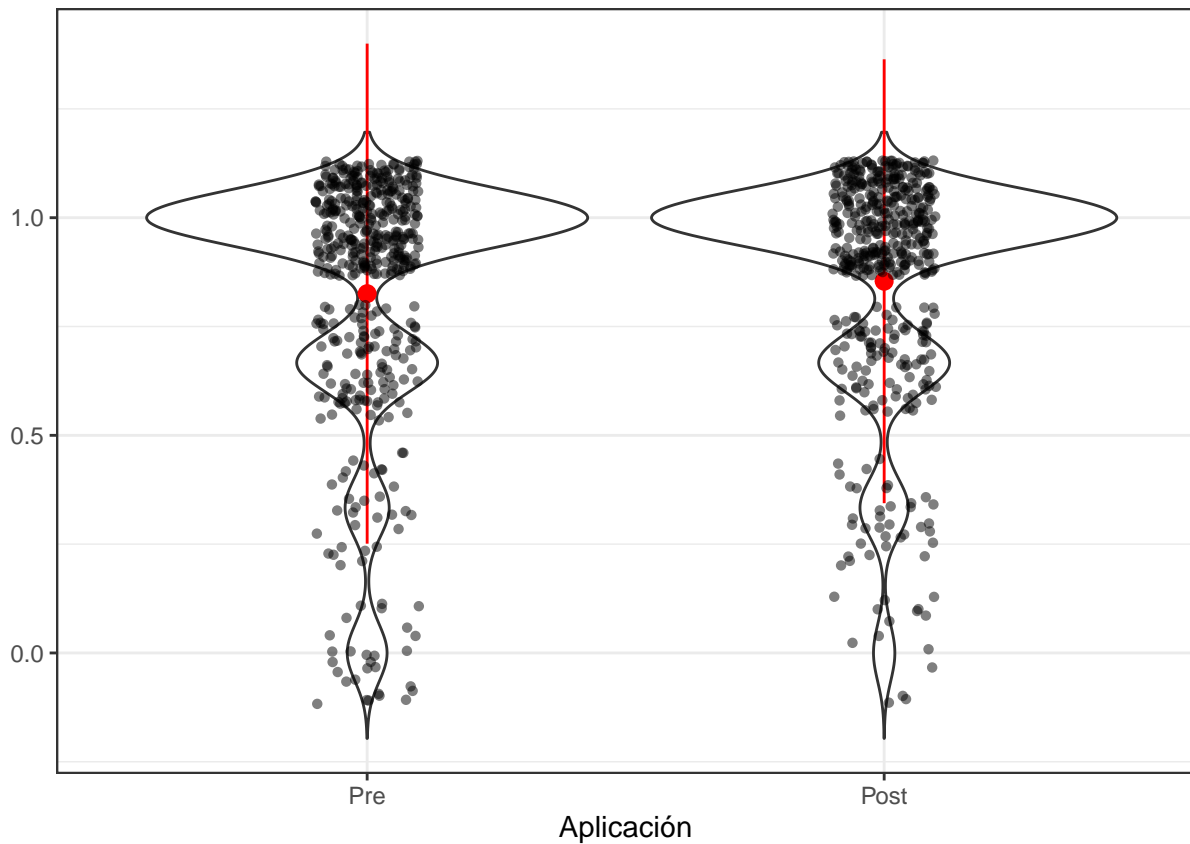
```
comparacion =  
  wilcox.test(  
    x      = pre_post$Afectivo_pre,  
    y      = pre_post$Afectivo_post,  
    alternative = "two.sided",  
    mu      = 0,  
    var.equal = TRUE,  
    paired   = TRUE,  
    conf.level = 0.95  
  )
```

comparacion

```
##  
## Wilcoxon signed rank test with continuity correction  
##  
## data: pre_post$Afectivo_pre and pre_post$Afectivo_post  
## V = 6980, p-value = 0.068  
## 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/primer_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: 0.10472 (negligible)
## 95 percent confidence interval:
##      lower      upper
## -0.0002917  0.2097391
```

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 = 0.45, 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 = 0.46, 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	468	0.87322	0.26371	1	0.94326	0	0	1	1
Cognitivo_post	468	0.89174	0.23728	1	0.95124	0	0	1	1

Comparación de medias

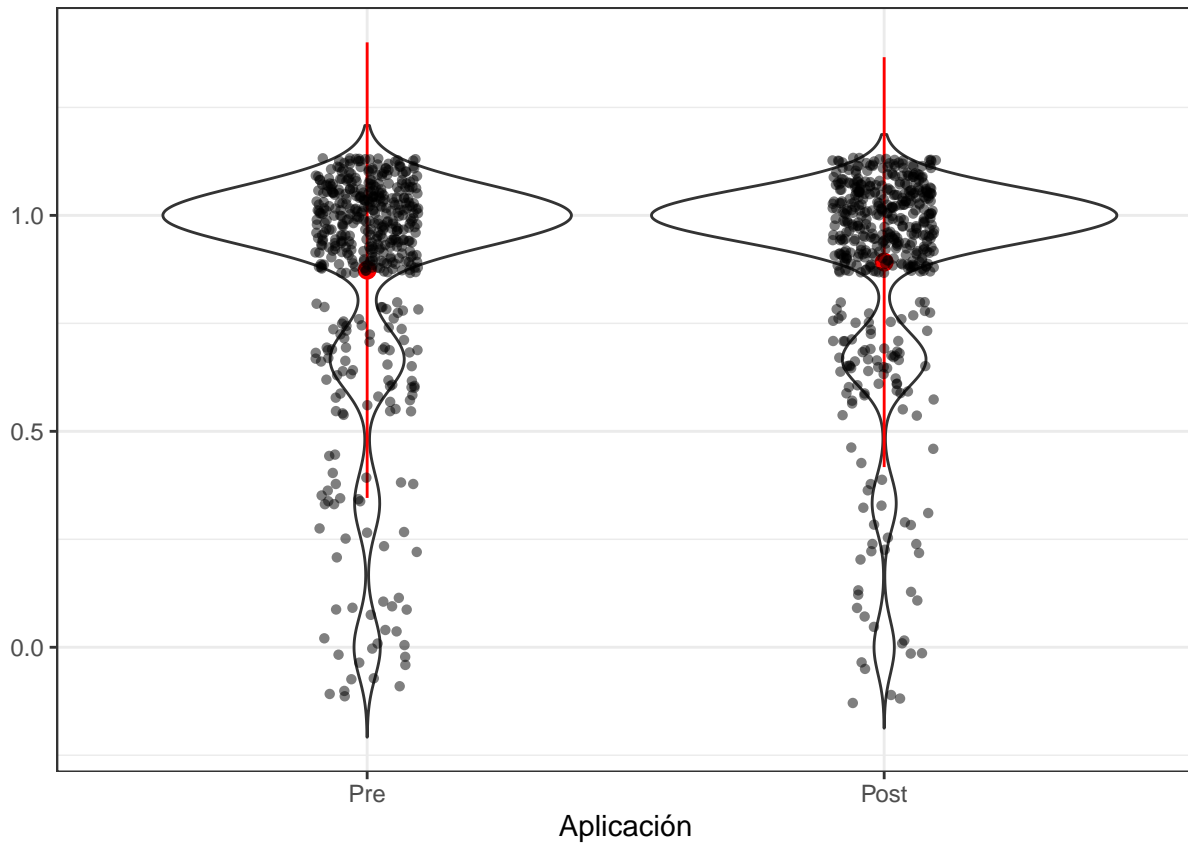
```
comparacion =
  wilcox.test(
    x      = pre_post$Cognitivo_pre,
    y      = pre_post$Cognitivo_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )
```

```
comparacion
```

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Cognitivo_pre and pre_post$Cognitivo_post
## V = 4249, p-value = 0.29
## 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/primer_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: 0.073709 (negligible)  
## 95 percent confidence interval:  
##   lower   upper  
## -0.02891 0.17633
```

```
# 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 = 0.424, 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 = 0.435, 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	468	0.84046	0.29054	1	0.90780	0	0	1	1
Conativo_post	468	0.86823	0.25433	1	0.93174	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
```

```

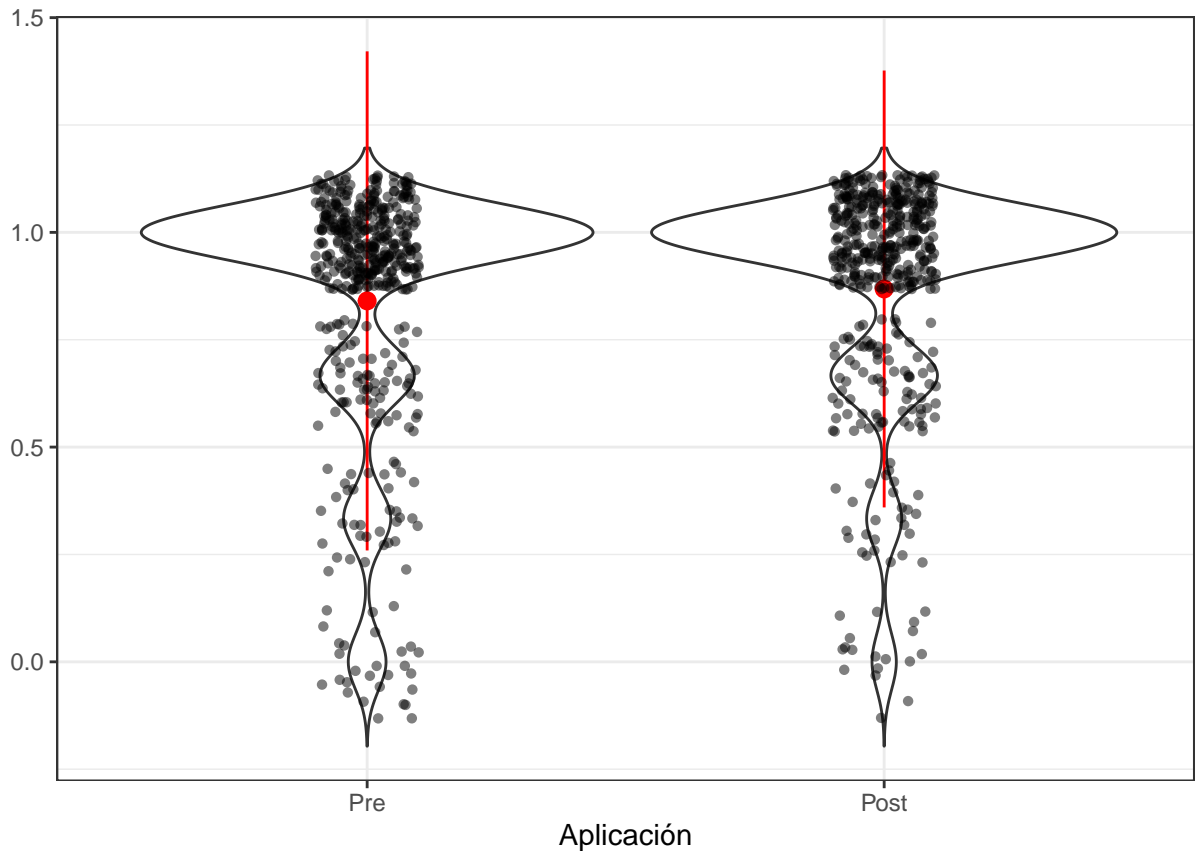
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Conativo_pre and pre_post$Conativo_post
## V = 4972, p-value = 0.052
## alternative hypothesis: true location shift is not equal to 0

```

```

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/primer_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: 0.10146 (negligible)  
## 95 percent confidence interval:  
##      lower      upper  
## 0.00029882 0.20262527
```

```
# 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[,1:18]

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

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

motivacion_pre[,7:18] = apply(motivacion_pre[,7:18], 2,
  function(x) {ifelse(x == "A", 1, 0)})

# 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")

calificacion = function(data, claves){
  for(i in 1:dim(data)[2]){
    data[,i] = apply(data[,i], 1,
      function(x) {ifelse(x == claves[i], 1, 0)})
  }
}

```

```

}
data
}

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

# Calificación

# General

motivacion_pre$Total_pre = apply(motivacion_pre[,7:18], 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(7,10,13,16)], 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(8,11,14,17)], 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$OrientacionResultado_pre = 4 - motivacion_pre$OrientacionAprendizaje_pre
motivacion_post$OrientacionResultado_post = 3 - motivacion_post$OrientacionAprendizaje_post

# Atribución interna

motivacion_pre$AtribucionInterna_pre = apply(motivacion_pre[,c(15,18)], 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(9,12)], 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",
                 "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")

unique(pivot_to_bind$Prueba)

## [1] "Total"          "Interes"          "OrientacionResultado" "AtribucionInteres"
## [5] "AtribucionExterna" "Expectativa"      "OrientacionAprendizaje"

pivot_to_bind$Prueba = str_replace_all(pivot_to_bind$Prueba,
                                         c("Total" = "Motivacion - Total",
                                             "Interes" = "Motivacion - Interes",
                                             "OrientacionResultado" = "Motivacion - Metas",
                                             "AtribucionExterna" = "Motivacion - Atribucion externa",
                                             "Expectativa" = "Motivacion - Expectativas"))

pivot_final = rbind(pivot_final, pivot_to_bind)

```

3.2.1 Alpha

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

3.2.1.1 Total pre

```

## [1] 0.79369

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

## [1] "El índice de confiabilidad cambia a 0.782468757023904 al eliminar el ítem 1"
## [1] "El índice de confiabilidad cambia a 0.78819576622514 al eliminar el ítem 2"
## [1] "El índice de confiabilidad cambia a 0.777242723784103 al eliminar el ítem 3"
## [1] "El índice de confiabilidad cambia a 0.777254987885137 al eliminar el ítem 4"
## [1] "El índice de confiabilidad cambia a 0.774370799725118 al eliminar el ítem 5"
## [1] "El índice de confiabilidad cambia a 0.775519749049695 al eliminar el ítem 6"
## [1] "El índice de confiabilidad cambia a 0.777894516973956 al eliminar el ítem 7"
## [1] "El índice de confiabilidad cambia a 0.779398242441121 al eliminar el ítem 8"
## [1] "El índice de confiabilidad cambia a 0.782217463765396 al eliminar el ítem 9"
## [1] "El índice de confiabilidad cambia a 0.771353650494053 al eliminar el ítem 10"
## [1] "El índice de confiabilidad cambia a 0.786498739766338 al eliminar el ítem 11"
## [1] "El índice de confiabilidad cambia a 0.777066074058406 al eliminar el ítem 12"

```

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

3.2.1.2 Total post

```
## [1] 0.61932
```

```
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 0.57606144905541 al eliminar el ítem 1"  
## [1] "El índice de confiabilidad cambia a 0.590121273884408 al eliminar el ítem 2"  
## [1] "El índice de confiabilidad cambia a 0.613084310423595 al eliminar el ítem 3"  
## [1] "El índice de confiabilidad cambia a 0.563507724352878 al eliminar el ítem 4"  
## [1] "El índice de confiabilidad cambia a 0.596604790923643 al eliminar el ítem 5"  
## [1] "El índice de confiabilidad cambia a 0.587063919017499 al eliminar el ítem 6"  
## [1] "El índice de confiabilidad cambia a 0.565355008206158 al eliminar el ítem 7"  
## [1] "El índice de confiabilidad cambia a 0.567618138091583 al eliminar el ítem 8"  
## [1] "El índice de confiabilidad cambia a 0.628365325537253 al eliminar el ítem 9"  
## [1] "El índice de confiabilidad cambia a 0.64797584555563 al eliminar el ítem 10"
```

```
# Amacemos los valores
```

```
alfa = rbind(alfa,  
             c("Motivación pre",  
               alpha(motivacion_pre[,7:18])))  
  
alfa = rbind(alfa,  
             c("Motivación post",  
               alpha(motivacion_post[,7:16])))
```

```
alpha(motivacion_pre[,c("1", "4", "7", "10")])
```

3.2.1.3 Interés pre

```
## [1] 0.66417
```

```
for(i in seq(length(colnames(motivacion_pre[,c("1", "4", "7", "10")])))){  
  x = alpha(motivacion_pre[,c("1", "4", "7", "10")][,-i])  
  print(paste("El índice de confiabilidad cambia a", x,  
              "al eliminar el ítem",  
              colnames(motivacion_pre[,c("1", "4", "7", "10")])[i]))  
}
```

```
## [1] "El índice de confiabilidad cambia a 0.628148720424915 al eliminar el ítem 1"  
## [1] "El índice de confiabilidad cambia a 0.58778116368899 al eliminar el ítem 4"
```

```
## [1] "El índice de confiabilidad cambia a 0.57936535648069 al eliminar el ítem 7"
## [1] "El índice de confiabilidad cambia a 0.597143818187136 al eliminar el ítem 10"
```

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

3.2.1.4 Interés post

```
## [1] 0.66124
```

```
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 0.653650308499103 al eliminar el ítem 1"
## [1] "El índice de confiabilidad cambia a 0.506022640049801 al eliminar el ítem 4"
## [1] "El índice de confiabilidad cambia a 0.541195957862625 al eliminar el ítem 7"
```

```
alpha(motivacion_pre[,c("2","5","8","11")])
```

3.2.1.5 Metas pre

```
## [1] 0.55185
```

```
for(i in seq(length(colnames(motivacion_pre[,c("2","5","8","11")])))){
  x = alpha(motivacion_pre[,c("2","5","8","11")][,-i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(motivacion_pre[,c("2","5","8","11")])[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.498833121907289 al eliminar el ítem 2"
## [1] "El índice de confiabilidad cambia a 0.486262273143154 al eliminar el ítem 5"
## [1] "El índice de confiabilidad cambia a 0.464445929493966 al eliminar el ítem 8"
## [1] "El índice de confiabilidad cambia a 0.470984867049863 al eliminar el ítem 11"
```

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

3.2.1.6 Metas post

```
## [1] 0.28162
```

```
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 0.201963896933784 al eliminar el ítem 2"
## [1] "El índice de confiabilidad cambia a 0.32108343696875 al eliminar el ítem 5"
## [1] "El índice de confiabilidad cambia a 0.0621751167934566 al eliminar el ítem 9"
```

3.2.1.7 Expectativas Este sólo está en el pre, ya que en el post la escala se subdividió

```
alpha(motivacion_pre[,c("3","6","9","12")])
```

```
## [1] 0.60279
```

```
for(i in seq(length(colnames(motivacion_pre[,c("3","6","9","12")])))){
  x = alpha(motivacion_pre[,c("3","6","9","12")][,-i])
  print(paste("El índice de confiabilidad cambia a", x,
              "al eliminar el ítem",
              colnames(motivacion_pre[,c("3","6","9","12")])[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.525780524891882 al eliminar el ítem 3"
## [1] "El índice de confiabilidad cambia a 0.513256556391975 al eliminar el ítem 6"
## [1] "El índice de confiabilidad cambia a 0.550812435675047 al eliminar el ítem 9"
## [1] "El índice de confiabilidad cambia a 0.54186286050618 al eliminar el ítem 12"
```

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

3.2.1.8 Atribución interna

```
## [1] 0.10725
```

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

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

3.2.1.9 Expectativa positiva

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

3.2.2 Indicadores psicométricos

```
analitem = item.exam(motivacion_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())

```

3.2.2.1 Total pre

Sample.SD	Item.total	Item.Tot.woi	Difficulty	Discrimination	Item.Reliab
0.346	0.516	0.399	0.862	0.321	0.178
0.462	0.536	0.377	0.691	0.603	0.248
0.337	0.564	0.456	0.870	0.326	0.190
0.269	0.566	0.482	0.921	0.217	0.152
0.312	0.591	0.496	0.890	0.304	0.185
0.326	0.579	0.478	0.880	0.310	0.188
0.279	0.556	0.468	0.915	0.220	0.155
0.390	0.560	0.433	0.813	0.465	0.219
0.388	0.536	0.406	0.815	0.429	0.208
0.310	0.622	0.533	0.892	0.310	0.193
0.449	0.539	0.386	0.719	0.590	0.242
0.418	0.590	0.458	0.775	0.535	0.246

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
0.366	0.534	0.378	0.840	0.353	0.196
0.427	0.498	0.306	0.761	0.488	0.213
0.462	0.438	0.217	0.692	0.485	0.202
0.317	0.592	0.469	0.886	0.294	0.188
0.431	0.477	0.279	0.754	0.444	0.206
0.366	0.489	0.325	0.841	0.309	0.179
0.321	0.582	0.456	0.883	0.300	0.187
0.379	0.568	0.412	0.827	0.400	0.215
0.492	0.409	0.168	0.592	0.526	0.201
0.475	0.322	0.081	0.658	0.376	0.153

```

# 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 = 0.225, 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 = 0.191, 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	882	0.83796	0.20215	0.91667	0.87618	0.12355	0	1	1
Total_post	882	0.77948	0.19106	0.80000	0.80340	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

```

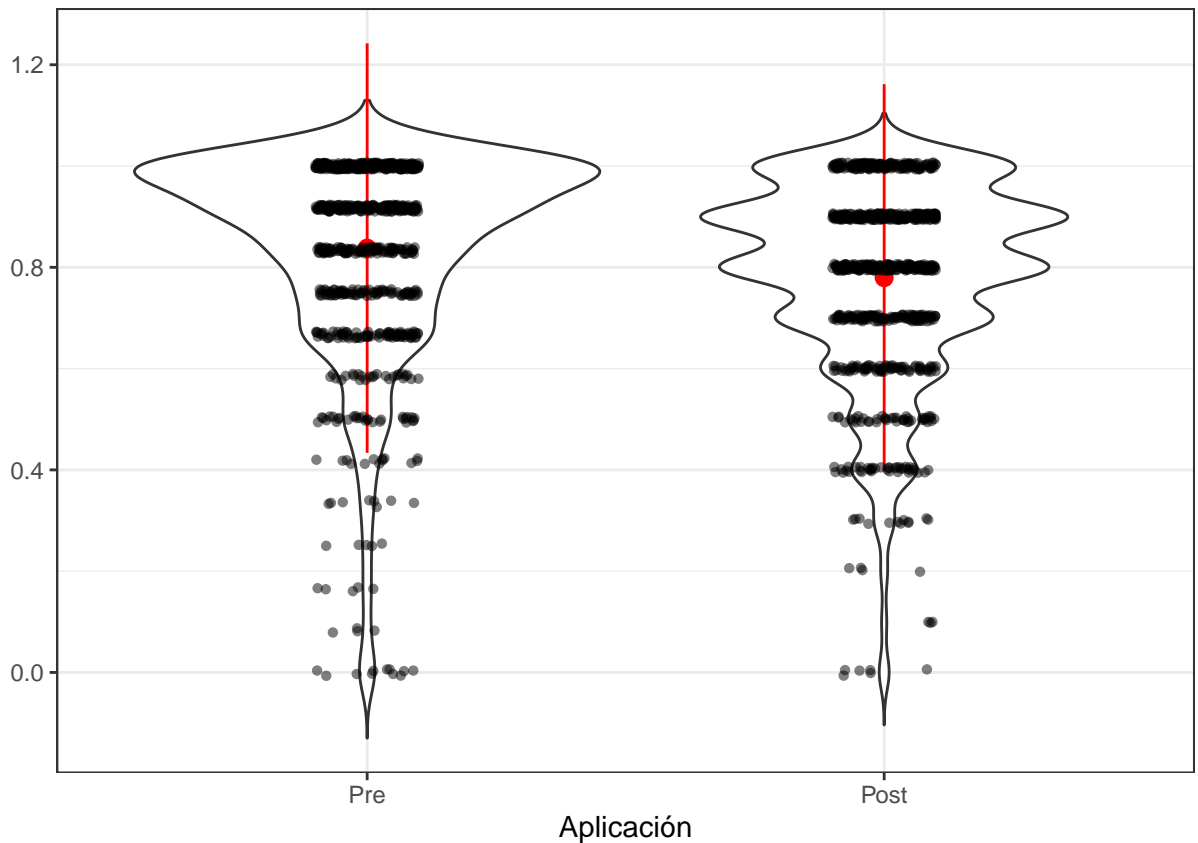
3.2.3.1.3 Comparación de medias

```

##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Total_pre and pre_post$Total_post
## V = 213576, 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/primer_aaplicacion/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: -0.29727 (small)
## 95 percent confidence interval:
## lower upper
## -0.37855 -0.21598
```

```
# 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 = 0.442, 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 = 0.442, 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	882	0.90108	0.21223	1	0.95715	0	0	1	1
Interes_post	882	0.87491	0.25281	1	0.93862	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 = 36950, p-value = 0.00012
## alternative hypothesis: true location shift is not equal to 0

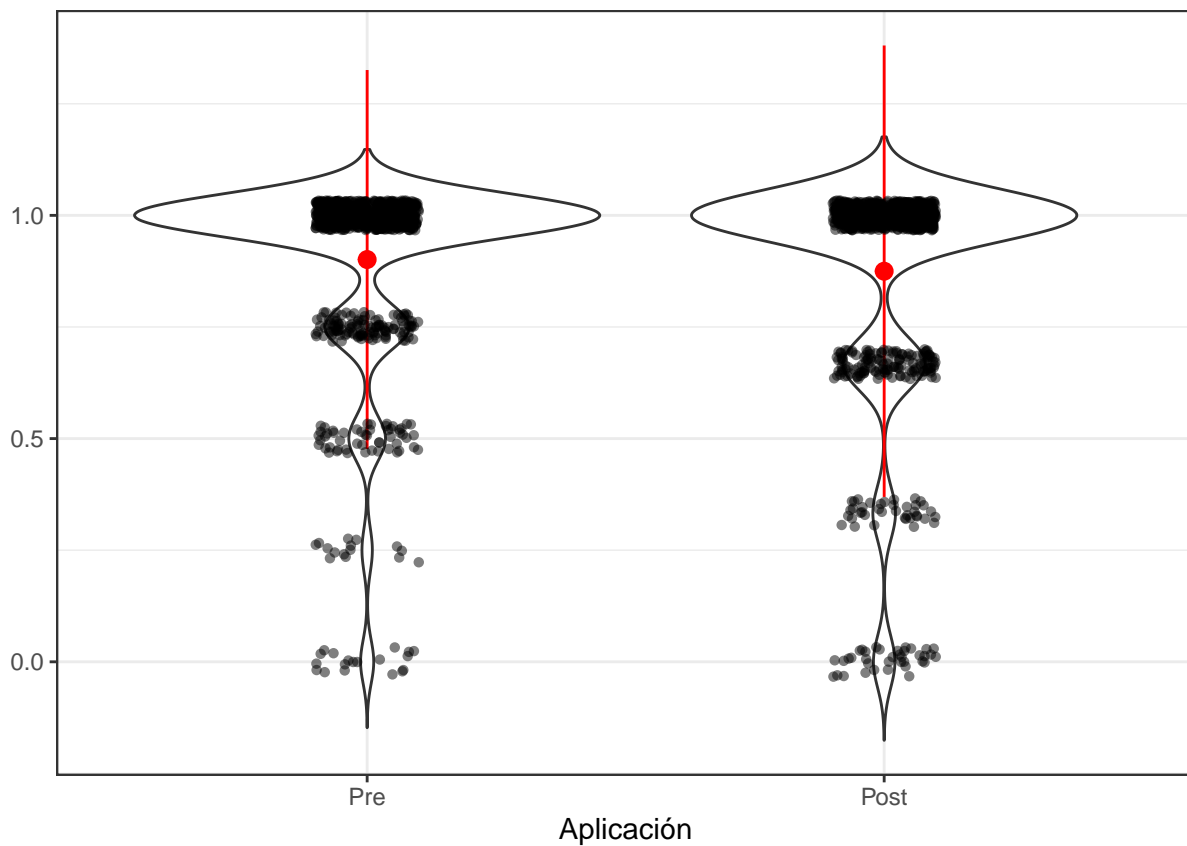
```

```

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/primer_aaplicacion/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: -0.11185 (negligible)
## 95 percent confidence interval:
##   lower   upper
## -0.19306 -0.03064
```

```

# 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

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

3.2.3.3.1 Estadísticos de normalidad

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

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

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$OrientacionResultado_pre
## D = 0.281, p-value <0.0000000000000002
```

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

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

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

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  pre_post$OrientacionResultado_post
## D = 0.24, 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("OrientacionResultado_pre",
                                                    "OrientacionResultado_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("OrientacionResultado_pre", "OrientacionResultado_post")

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

3.2.3.3.2 Descriptivos

vars	n	mean	sd	median	trimmed	mad	min	max	range
OrientacionResultado_pre	882	0.22137	0.26866	0.25000	0.17528	0.37065	0	1	1
OrientacionResultado_post	882	0.29327	0.28791	0.33333	0.26440	0.49420	0	1	1

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

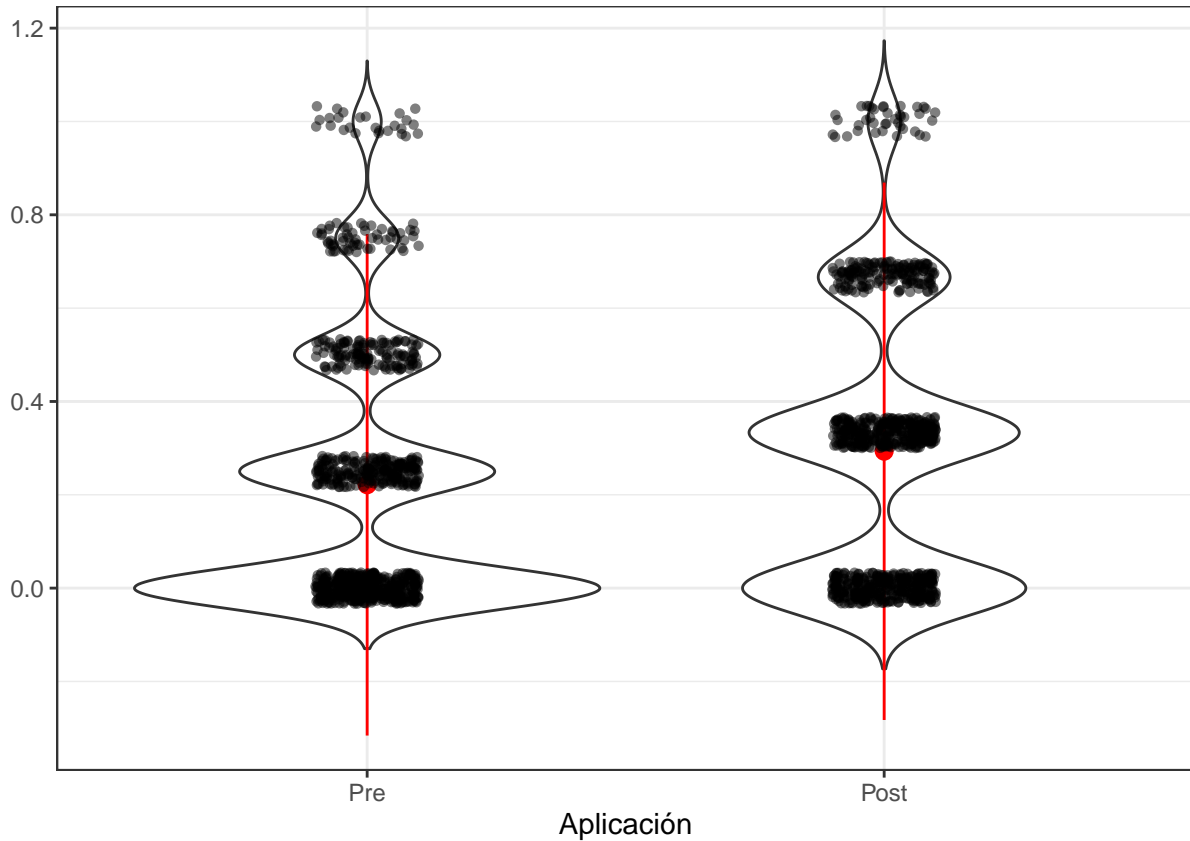
```
comparacion
```

3.2.3.3.3 Comparación de medias

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$OrientacionResultado_pre and pre_post$OrientacionResultado_post
## V = 86774, p-value = 0.000000036
## alternative hypothesis: true location shift is not equal to 0
```

```
pre_post %>%
  dplyr::select(OrientacionResultado_pre, OrientacionResultado_post) %>%
  pivot_longer(cols = c(OrientacionResultado_pre, OrientacionResultado_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/primer_aaplicacion/motivacion_metas_resultado.png")
```

```
size_effect =
  cohen.d(pre_post$OrientacionResultado_post,
          pre_post$OrientacionResultado_pre, paired = TRUE)

size_effect
```

3.2.3.3.4 Tamaño del efecto

```
##
## Cohen's d
##
## d estimate: 0.25815 (small)
## 95 percent confidence interval:
```



```
## lower upper
## 0.17247 0.34382

# 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)
```

```
pre_post$AtribucionExterna_pre = 1 - pre_post$AtribucionInterna_pre
pre_post$AtribucionExterna_post = 1 - pre_post$AtribucionInterna_post
```

3.2.3.4 Atribución externa

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

3.2.3.4.1 Estadísticos de normalidad

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

##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: pre_post$AtribucionExterna_pre
## D = 0.416, p-value <0.0000000000000002
print("Estadístico de normalidad post")

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

##
```

```
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: pre_post$AtribucionExterna_post
## D = 0.308, 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("AtribucionExterna_pre",
                                                    "AtribucionExterna_post")])),
                  check.names = FALSE, stringsAsFactors = FALSE)

summ$vars = c("AtribucionExterna_pre", "AtribucionExterna_post")

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

3.2.3.4.2 Descriptivos

vars	n	mean	sd	median	trimmed	mad	min	max	range
AtribucionExterna_pre	882	0.20238	0.31984	0.0	0.13810	0.0000	0	1	1
AtribucionExterna_post	882	0.31179	0.33650	0.5	0.26487	0.7413	0	1	1

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

comparacion
```

3.2.3.4.3 Comparación de medias

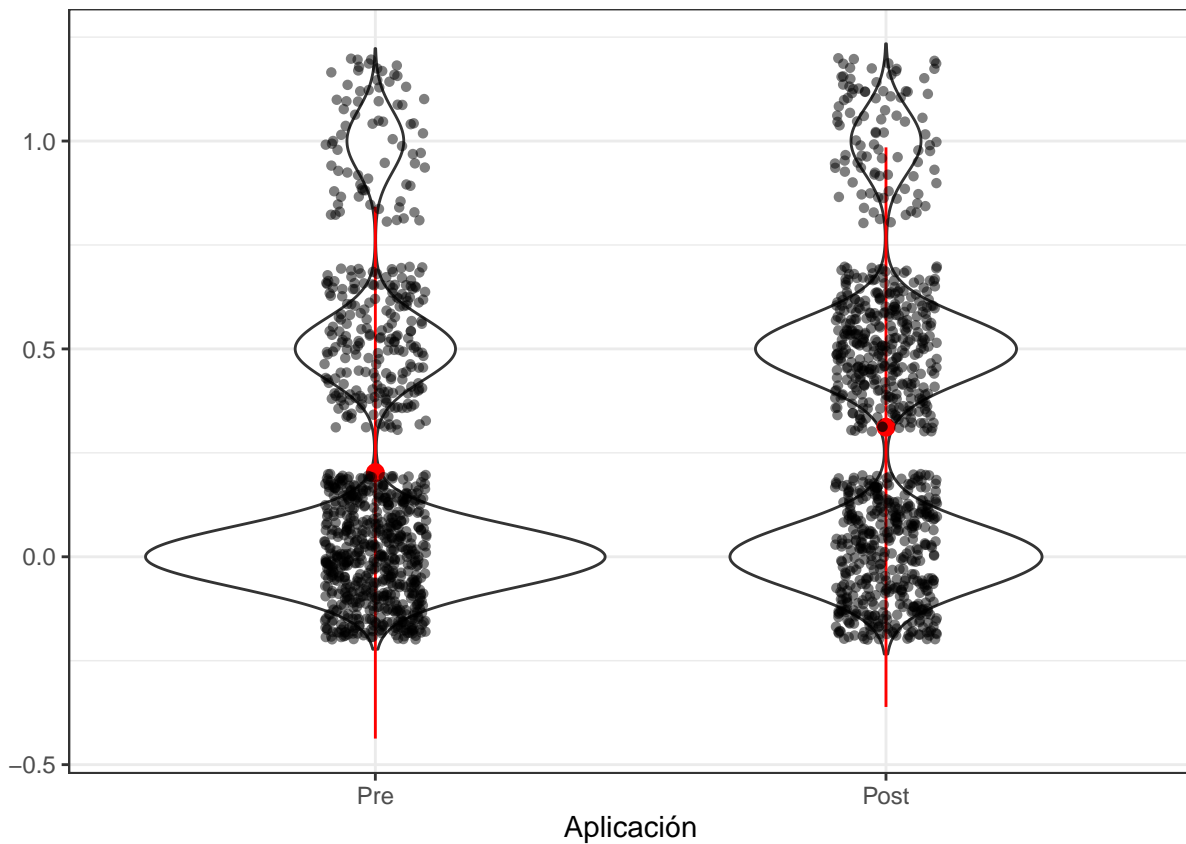
```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$AtribucionExterna_pre and pre_post$AtribucionExterna_post
## V = 34057, p-value = 0.000000000000071
## alternative hypothesis: true location shift is not equal to 0

pre_post %>%
  dplyr::select(AtribucionExterna_pre, AtribucionExterna_post) %>%
  pivot_longer(cols = c(AtribucionExterna_pre, AtribucionExterna_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/primer_aplicacion/motivacion_AtribucionExterna.png")

```

```

size_effect =
  cohen.d(pre_post$AtribucionExterna_post,
    pre_post$AtribucionExterna_pre, paired = TRUE)

size_effect

```

3.2.3.4.4 Tamaño del efecto

```
##
## Cohen's d
##
## d estimate: 0.33325 (small)
## 95 percent confidence interval:
##   lower   upper
## 0.24417 0.42234

# Separamos los descriptivos importantes
desc = summ[,1:10] %>%
  dplyr::select(c("n", "mean", "sd", "min", "max")) %>%
  mutate(Prueba = "Motivación - Atribución externa") %>%
  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 externa",
    `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 = 0.477, 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 = 0.455, 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	882	0.87075	0.27426	1	0.93484	0	0	1	1
Expectativa_post	882	0.83673	0.30876	1	0.90864	0	0	1	1

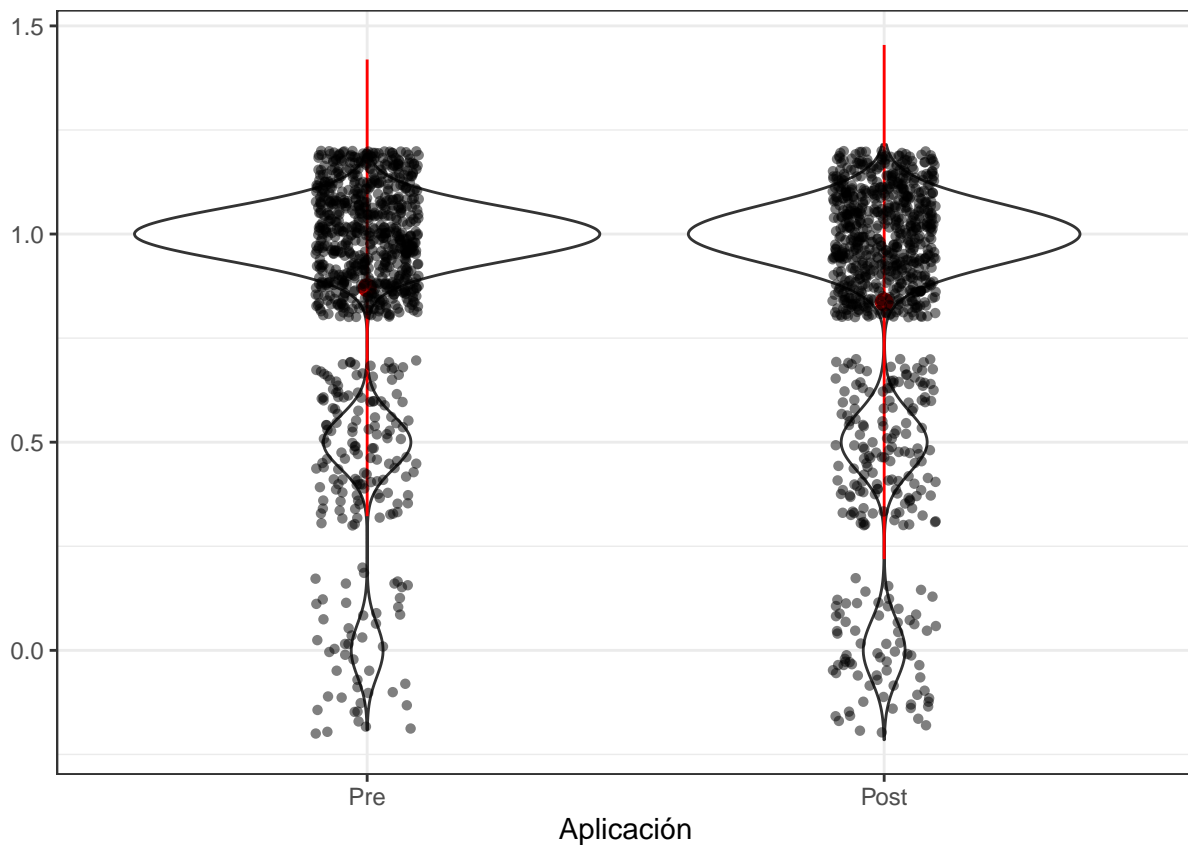
```
comparacion =
  wilcox.test(
    x      = pre_post$Expectativa_pre,
    y      = pre_post$Expectativa_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )

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 = 22758, p-value = 0.0081
## 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/primer_aaplicacion/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: -0.11639 (negligible)
## 95 percent confidence interval:
##      lower      upper
## -0.20145 -0.03133

# 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(Vaca),
  Vaca != '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", "Pantera")

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(Burro),
                        Burro != 'NULL')

item_1_post = c("Caballo", "Gallina", "Conejo",
                "Leopardo", "Ardilla", "Cerdo", "Cangrejo", "Burro")

item_2_post = c("Estrella", "Gato", "Abeja",
                "Cebra", "Rana", "Armadillo", "Tortuga", "Iguana")

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

mem_audi_post[,7:30] = apply(mem_audi_post[,7:30], 2,
                             function(x) str_replace_all(x, 'NULL', '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] 0.70579
```

```
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 0.666130900139336 al eliminar el ítem Caballo"  
## [1] "El índice de confiabilidad cambia a 0.683716238977905 al eliminar el ítem Perro"  
## [1] "El índice de confiabilidad cambia a 0.665717639419595 al eliminar el ítem Aguila"  
## [1] "El índice de confiabilidad cambia a 0.661316235704946 al eliminar el ítem Pollito"  
## [1] "El índice de confiabilidad cambia a 0.674909980891176 al eliminar el ítem Foca"  
## [1] "El índice de confiabilidad cambia a 0.693572310541209 al eliminar el ítem Vaca"  
## [1] "El índice de confiabilidad cambia a 0.676972198304777 al eliminar el ítem Cocodrilo"  
## [1] "El índice de confiabilidad cambia a 0.695434484575072 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] 0.69164
```

```
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 0.675157114681925 al eliminar el ítem Oveja"  
## [1] "El índice de confiabilidad cambia a 0.661596394789176 al eliminar el ítem Elefante"  
## [1] "El índice de confiabilidad cambia a 0.660489487644126 al eliminar el ítem Tiburon"  
## [1] "El índice de confiabilidad cambia a 0.655202419155764 al eliminar el ítem Caracol"  
## [1] "El índice de confiabilidad cambia a 0.65212140477163 al eliminar el ítem Ratón"  
## [1] "El índice de confiabilidad cambia a 0.671070396646643 al eliminar el ítem Gato"
```

```
## [1] "El índice de confiabilidad cambia a 0.656589257505955 al eliminar el ítem Tortuga"
## [1] "El índice de confiabilidad cambia a 0.666824760386176 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] 0.67687
```

```
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 0.650752844569212 al eliminar el ítem Caballo"
## [1] "El índice de confiabilidad cambia a 0.640724637444536 al eliminar el ítem Gallina"
## [1] "El índice de confiabilidad cambia a 0.652021815860059 al eliminar el ítem Conejo"
## [1] "El índice de confiabilidad cambia a 0.630418362658404 al eliminar el ítem Leopardo"
## [1] "El índice de confiabilidad cambia a 0.632130170859231 al eliminar el ítem Ardilla"
## [1] "El índice de confiabilidad cambia a 0.641881600635921 al eliminar el ítem Cerdo"
## [1] "El índice de confiabilidad cambia a 0.63723377816997 al eliminar el ítem Cangrejo"
## [1] "El índice de confiabilidad cambia a 0.687275043556359 al eliminar el ítem Burro"
```

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] 0.56339
```

```
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 0.531445147939311 al eliminar el ítem Estrella"
## [1] "El índice de confiabilidad cambia a 0.535341852298021 al eliminar el ítem Gato"
## [1] "El índice de confiabilidad cambia a 0.53116985218513 al eliminar el ítem Abeja"
## [1] "El índice de confiabilidad cambia a 0.500245788573356 al eliminar el ítem Cebra"
## [1] "El índice de confiabilidad cambia a 0.535832566289839 al eliminar el ítem Rana"
## [1] "El índice de confiabilidad cambia a 0.523945049294061 al eliminar el ítem Armadillo"
## [1] "El índice de confiabilidad cambia a 0.539197566024435 al eliminar el ítem Tortuga"
## [1] "El índice de confiabilidad cambia a 0.542139899120167 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
0.369	0.602	0.469	0.837	0.385	0.222
0.475	0.556	0.367	0.658	0.581	0.264

0.471	0.618	0.446	0.668	0.618	0.291
0.406	0.623	0.480	0.792	0.470	0.253
0.489	0.593	0.407	0.605	0.643	0.290
0.480	0.523	0.324	0.641	0.545	0.251
0.450	0.571	0.397	0.717	0.540	0.257
0.489	0.521	0.318	0.606	0.589	0.255

```
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
0.468	0.517	0.325	0.675	0.517	0.242
0.487	0.573	0.385	0.615	0.641	0.279
0.432	0.559	0.392	0.752	0.475	0.241
0.464	0.587	0.413	0.687	0.574	0.272
0.460	0.597	0.426	0.696	0.589	0.275
0.500	0.547	0.347	0.500	0.636	0.274
0.459	0.581	0.407	0.699	0.556	0.266

0.463	0.545	0.362	0.690	0.525	0.252
-------	-------	-------	-------	-------	-------

```
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
0.387	0.522	0.352	0.817	0.357	0.202
0.400	0.565	0.396	0.801	0.419	0.226
0.400	0.523	0.345	0.800	0.398	0.209
0.440	0.611	0.433	0.738	0.519	0.269
0.438	0.605	0.427	0.741	0.549	0.265
0.465	0.584	0.387	0.684	0.584	0.272
0.436	0.588	0.406	0.746	0.534	0.256
0.481	0.447	0.213	0.638	0.528	0.215

```
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
0.280	0.442	0.303	0.915	0.206	0.123
0.482	0.501	0.258	0.633	0.575	0.242
0.495	0.518	0.271	0.572	0.581	0.256
0.499	0.585	0.353	0.540	0.678	0.292
0.468	0.493	0.256	0.676	0.487	0.231
0.477	0.525	0.290	0.650	0.537	0.250
0.438	0.468	0.244	0.742	0.442	0.205
0.414	0.446	0.233	0.781	0.372	0.185

```
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 = 0.131, 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 = 0.113, 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	906	0.64480	0.23754	0.6875	0.66839	0.18532	0	1	1
Total_post	906	0.67777	0.20355	0.6875	0.69370	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
  )

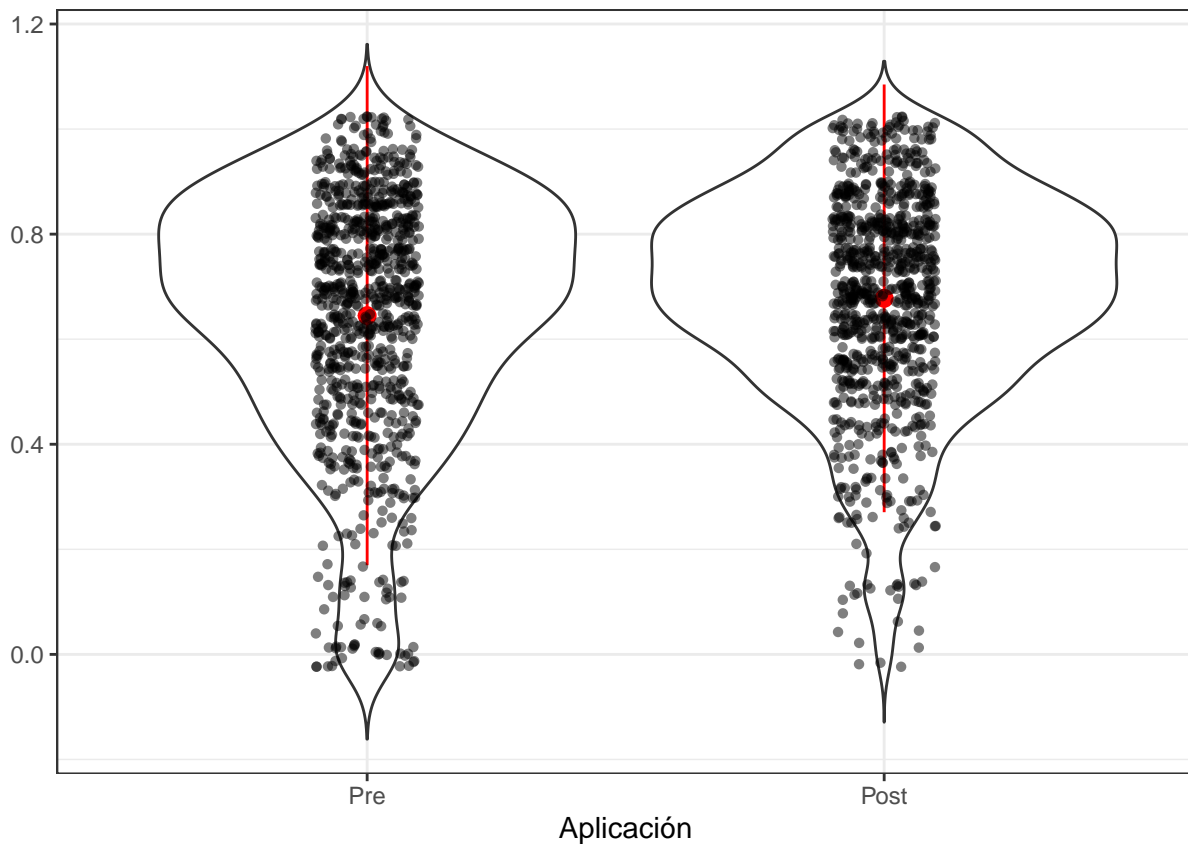
comparacion
```

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 = 132619, p-value = 0.00022
## 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/primer_aaplicacion/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: 0.14826 (negligible)
## 95 percent confidence interval:
##   lower   upper
## 0.081552 0.214975

# 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,][7:22])))
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[,7:22] = apply(mem_vis_pre[,7:22], 2,
                          function(x) str_replace_all(x,'NULL', '0'))

item_1_pre = c("Gym","Museo","Cajas")
item_2_pre = c("Salón","Aeropuerto","Iglesia")

incorrectos_1_pre = c("Cine",    "Lavadoras",    "Oficina")
incorrectos_2_pre = c("Sala",    "Paradero",    "Café")

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,][7:22])))
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),
        Iglesia != 'NULL')

mem_vis_post[,7:22] = apply(mem_vis_post[,7:22], 2,
        function(x) str_replace_all(x,'NULL', '0'))

item_1_post = c("Iglesia", "GYM", "Zoo")
item_2_post = c("Vet", "Playa", "Fruteria")

incorrectos_1_post = c("Cancha","Colegio_","Par. Diversiones")
incorrectos_2_post = c("Aeropuerto","Helado","Parque")

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 Ítem 1

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

```
alpha(data_item_1_pre)
```

```
## [1] 0.44137
```

```
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 0.328143644082117 al eliminar el ítem Gym"
```

```
## [1] "El índice de confiabilidad cambia a 0.237793787517137 al eliminar el ítem Museo"
```

```
## [1] "El índice de confiabilidad cambia a 0.45193975695175 al eliminar el ítem Cajas"
```

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] 0.54036
```

```
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 0.456399758403991 al eliminar el ítem Salón"
```

```
## [1] "El índice de confiabilidad cambia a 0.405150194030566 al eliminar el ítem Aeropuerto"
```

```
## [1] "El índice de confiabilidad cambia a 0.455489797679838 al eliminar el ítem Iglesia"
```

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] 0.46657
```

```
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 0.443070925825663 al eliminar el ítem Iglesia"
```

```
## [1] "El índice de confiabilidad cambia a 0.393225253630711 al eliminar el ítem GYM"
```

```
## [1] "El índice de confiabilidad cambia a 0.266102535230204 al eliminar el ítem Zoo"
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] 0.58181

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 0.557675244010648 al eliminar el ítem Vet"
## [1] "El índice de confiabilidad cambia a 0.453579110502133 al eliminar el ítem Playa"
## [1] "El índice de confiabilidad cambia a 0.406329929515591 al eliminar el ítem Fruteria"

# Amacemos 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
0.497	0.697	0.277	0.440	0.749	0.346
0.495	0.725	0.325	0.428	0.778	0.359
0.482	0.638	0.206	0.633	0.636	0.308

```

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
0.494	0.713	0.341	0.424	0.760	0.353
0.497	0.735	0.373	0.555	0.840	0.365
0.500	0.718	0.342	0.484	0.765	0.359

```

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
0.497	0.718	0.255	0.441	0.888	0.357
0.391	0.641	0.277	0.812	0.519	0.251
0.442	0.728	0.345	0.735	0.717	0.321

```
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
0.381	0.667	0.336	0.824	0.475	0.254
0.413	0.738	0.410	0.782	0.578	0.305
0.488	0.804	0.440	0.610	0.870	0.392

```
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 = 0.187, 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 = 0.157, 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	904	0.37795	0.29007	0.33333	0.35589	0.2471	0	1	1
Total_post	904	0.61486	0.29732	0.66667	0.63651	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

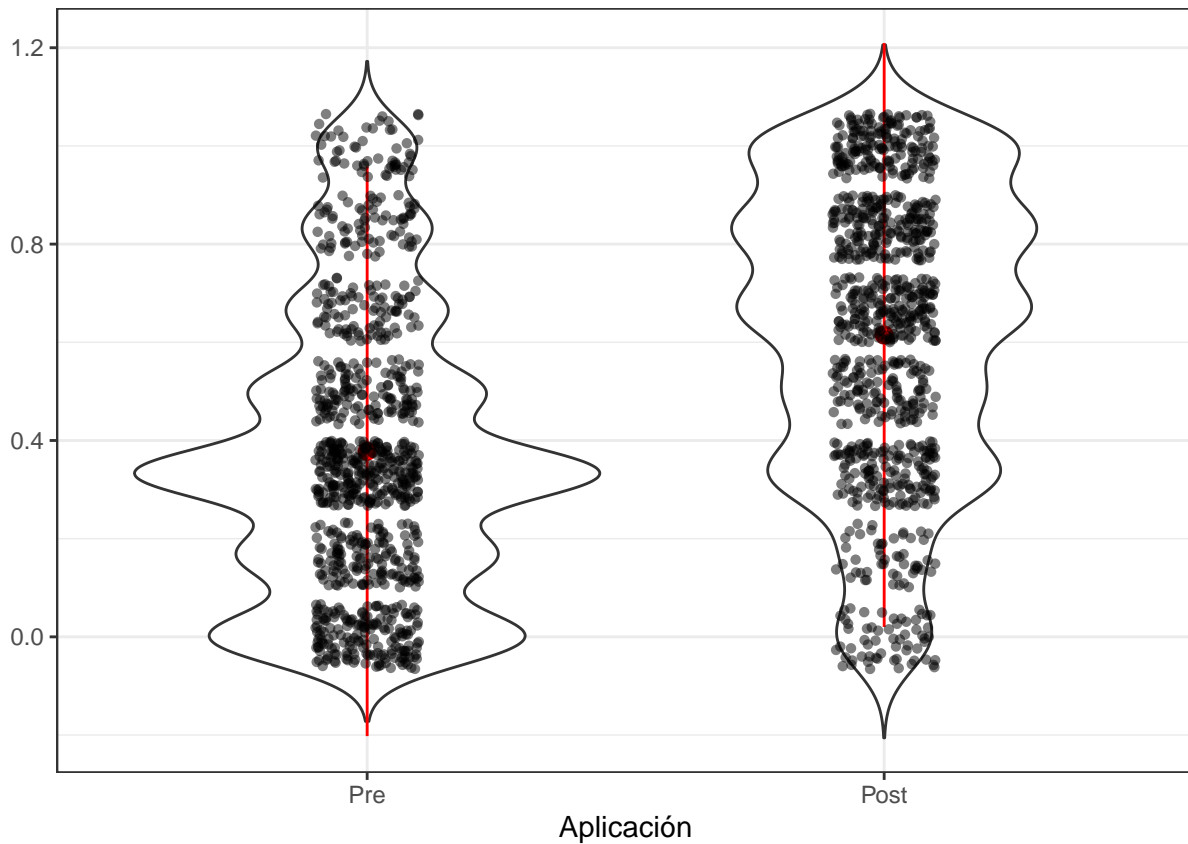
```

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  pre_post$Total_pre and pre_post$Total_post
## V = 37059, 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/primer_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: 0.80654 (large)
## 95 percent confidence interval:
##   lower   upper
## 0.71740 0.89569

# 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", 26), rep(1:2, each = 13),
  rep(paste("_", rep(1:13, 2))))

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

inhibicion_pre = inhibicion_pre[,1:32]

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', 'N', 'M', 'N',
                          'M', 'N', 'M', 'M', 'N', 'M', 'N',
                          'M', 'N', 'M', 'N', 'M', 'M', 'M', 'N',
                          'M', 'N', 'N', 'M')

inhibicion_pre[,7:32] = calificacion(inhibicion_pre[,7:32], claves_inhibicion_pre)

inhibicion_pre$`Aciertos item 1_pre` = apply(inhibicion_pre[,7:19], 1,
                                              function(x) sum(x, na.rm= FALSE))

inhibicion_pre$`Aciertos item 2_pre` = apply(inhibicion_pre[,20:32], 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[,20:32], 2, function(x) as.numeric(x))
```

```
alpha(data_item_2_pre)
```

3.3.3.1.1 Pre

```
## [1] 0.94683
```

```
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 0.943668938756567 al eliminar el ítem Item 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.942311085780618 al eliminar el ítem Item 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.942862514006446 al eliminar el ítem Item 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.942043997226524 al eliminar el ítem Item 2 _ 4"
## [1] "El índice de confiabilidad cambia a 0.941565185985918 al eliminar el ítem Item 2 _ 5"
## [1] "El índice de confiabilidad cambia a 0.94357406532119 al eliminar el ítem Item 2 _ 6"
## [1] "El índice de confiabilidad cambia a 0.9438365904305 al eliminar el ítem Item 2 _ 7"
## [1] "El índice de confiabilidad cambia a 0.941726679095834 al eliminar el ítem Item 2 _ 8"
## [1] "El índice de confiabilidad cambia a 0.941783542237724 al eliminar el ítem Item 2 _ 9"
## [1] "El índice de confiabilidad cambia a 0.942442668802986 al eliminar el ítem Item 2 _ 10"
## [1] "El índice de confiabilidad cambia a 0.942149512168751 al eliminar el ítem Item 2 _ 11"
## [1] "El índice de confiabilidad cambia a 0.943359453315759 al eliminar el ítem Item 2 _ 12"
## [1] "El índice de confiabilidad cambia a 0.943118500338867 al eliminar el ítem Item 2 _ 13"
```

```
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] 0.92799
```

```
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 0.924629490072192 al eliminar el ítem Item 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.924848780693211 al eliminar el ítem Item 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.923773214924156 al eliminar el ítem Item 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.917028514333901 al eliminar el ítem Item 2 _ 4"
## [1] "El índice de confiabilidad cambia a 0.917055094202674 al eliminar el ítem Item 2 _ 5"
## [1] "El índice de confiabilidad cambia a 0.917900673958554 al eliminar el ítem Item 2 _ 6"
## [1] "El índice de confiabilidad cambia a 0.917381514055354 al eliminar el ítem Item 2 _ 7"
## [1] "El índice de confiabilidad cambia a 0.918948480902771 al eliminar el ítem Item 2 _ 8"
## [1] "El índice de confiabilidad cambia a 0.920233091823875 al eliminar el ítem Item 2 _ 9"
## [1] "El índice de confiabilidad cambia a 0.923912276572945 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
0.423	0.751	0.703	0.767	0.615	0.317
0.412	0.791	0.750	0.784	0.605	0.326
0.424	0.775	0.731	0.766	0.638	0.328
0.407	0.799	0.760	0.791	0.607	0.325
0.432	0.813	0.775	0.753	0.669	0.351
0.435	0.756	0.708	0.747	0.636	0.329
0.433	0.748	0.699	0.750	0.636	0.324
0.422	0.808	0.770	0.769	0.646	0.341
0.412	0.807	0.769	0.783	0.615	0.333
0.418	0.787	0.745	0.774	0.628	0.329
0.428	0.796	0.755	0.759	0.669	0.341
0.428	0.761	0.714	0.759	0.641	0.326
0.434	0.769	0.723	0.749	0.667	0.333

```
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
0.299	0.703	0.641	0.901	0.292	0.210
0.359	0.715	0.641	0.848	0.451	0.257
0.362	0.733	0.662	0.845	0.466	0.266
0.355	0.834	0.787	0.853	0.442	0.296
0.357	0.834	0.786	0.851	0.445	0.297
0.361	0.821	0.770	0.846	0.460	0.297
0.347	0.829	0.782	0.861	0.419	0.287
0.350	0.805	0.752	0.858	0.425	0.281
0.344	0.784	0.728	0.863	0.410	0.269
0.371	0.735	0.662	0.836	0.484	0.272

```
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 = 0.26, 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 = 0.369, 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	905	0.78436	0.31643	0.92308	0.85019	0.11405	0	1	1

Total_post	905	0.87094	0.25665	1.00000	0.93986	0.00000	0	1	1
------------	-----	---------	---------	---------	---------	---------	---	---	---

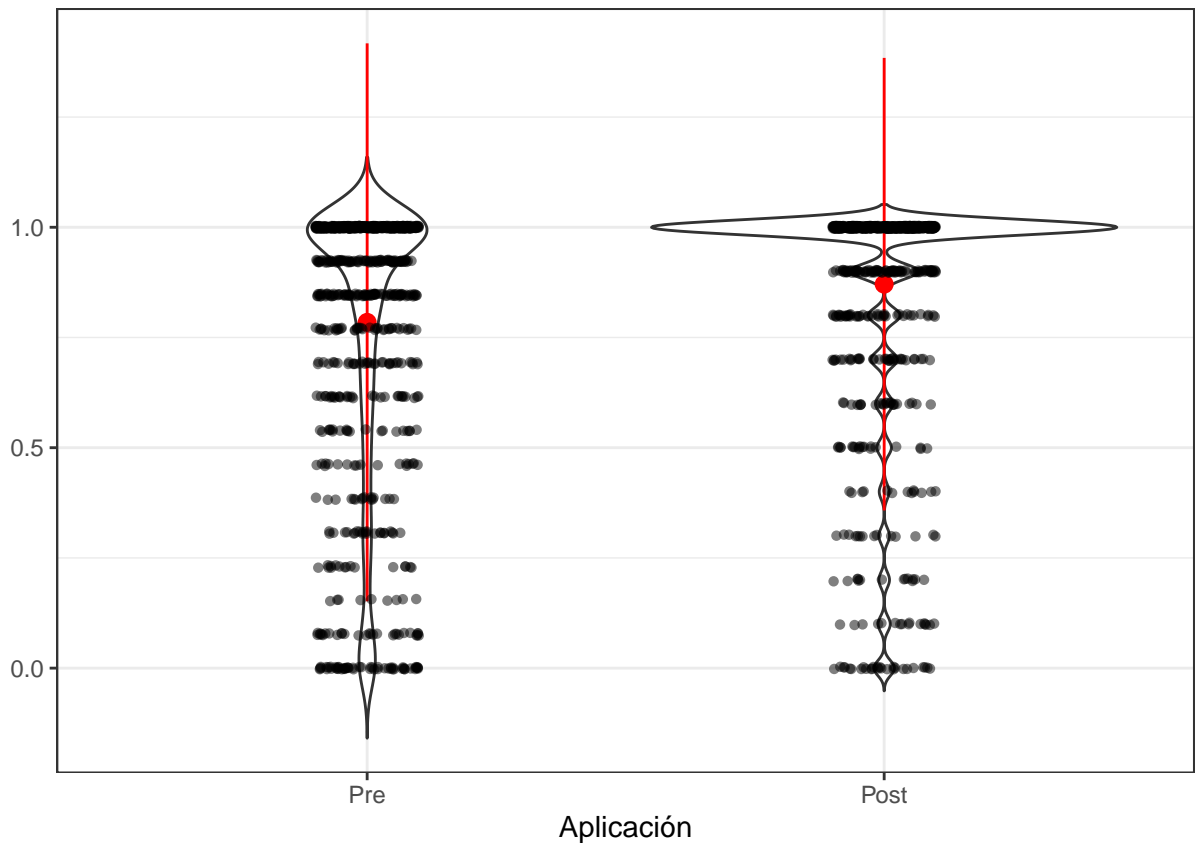
```
comparacion =
  wilcox.test(
    x      = pre_post$Total_pre,
    y      = pre_post$Total_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )

comparacion
```

3.3.3.3 Comparación de medias

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Total_pre and pre_post$Total_post
## V = 43540, p-value = 0.00000000000000041
## 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/primer_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: 0.29855 (small)
## 95 percent confidence interval:
##   lower   upper
## 0.22443 0.37267
```

```
# 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", 26), rep(1:2, each = 13),
  rep(paste("_", rep(1:13, 2))))

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

colnames(flexibilidad_pre) = vector

flexibilidad_pre = flexibilidad_pre[,1:32]

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

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', 'M', 'N', 'N', '3', 'M', 'M', '3',
  'N', 'N', 'M', '3', '3', 'M', 'N', 'M')

flexibilidad_pre[,7:32] = calificacion(flexibilidad_pre[,7:32],
  claves_flexibilidad_pre)

```

```

flexibilidad_pre$`Aciertos item 1_pre` = apply(flexibilidad_pre[,7:19], 1,
                                              function(x) sum(x, na.rm= FALSE))

flexibilidad_pre$`Aciertos item 2_pre` = apply(flexibilidad_pre[,20:32], 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`))

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

claves_flexibilidad_post = c('F', 'S', 'E', 'E', 'F', 'S', 'F', 'E', 'F', '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[,20:32])
```

3.3.4.1.1 Pre

```
## [1] 0.94877
```



```
for(i in seq(length(colnames(flexibilidad_pre[,20:32])))){
  x = alpha(flexibilidad_pre[,20:32][,-i])
  print(paste("El índice de confiabilidad cambia a", x,
             "al eliminar el ítem",
             colnames(flexibilidad_pre[,20:32])[i]))
}
```

```
## [1] "El índice de confiabilidad cambia a 0.94842931984324 al eliminar el ítem Item 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.947478057441881 al eliminar el ítem Item 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.944235134796501 al eliminar el ítem Item 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.943451702455763 al eliminar el ítem Item 2 _ 4"
## [1] "El índice de confiabilidad cambia a 0.946288728849321 al eliminar el ítem Item 2 _ 5"
## [1] "El índice de confiabilidad cambia a 0.943215546758642 al eliminar el ítem Item 2 _ 6"
## [1] "El índice de confiabilidad cambia a 0.942961254762305 al eliminar el ítem Item 2 _ 7"
## [1] "El índice de confiabilidad cambia a 0.942818334946887 al eliminar el ítem Item 2 _ 8"
## [1] "El índice de confiabilidad cambia a 0.945984105162767 al eliminar el ítem Item 2 _ 9"
## [1] "El índice de confiabilidad cambia a 0.94604114029321 al eliminar el ítem Item 2 _ 10"
## [1] "El índice de confiabilidad cambia a 0.943015273322182 al eliminar el ítem Item 2 _ 11"
## [1] "El índice de confiabilidad cambia a 0.943536476061499 al eliminar el ítem Item 2 _ 12"
## [1] "El índice de confiabilidad cambia a 0.943709314830116 al eliminar el ítem Item 2 _ 13"
```

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

3.3.4.1.2 Post

```
## [1] 0.97649
```

```
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 0.974158392849903 al eliminar el ítem Item 2 _ 1"
## [1] "El índice de confiabilidad cambia a 0.97434431540009 al eliminar el ítem Item 2 _ 2"
## [1] "El índice de confiabilidad cambia a 0.973316363440044 al eliminar el ítem Item 2 _ 3"
## [1] "El índice de confiabilidad cambia a 0.973682504054054 al eliminar el ítem Item 2 _ 4"
## [1] "El índice de confiabilidad cambia a 0.974733724646625 al eliminar el ítem Item 2 _ 5"
## [1] "El índice de confiabilidad cambia a 0.973571885201917 al eliminar el ítem Item 2 _ 6"
## [1] "El índice de confiabilidad cambia a 0.973549549730732 al eliminar el ítem Item 2 _ 7"
## [1] "El índice de confiabilidad cambia a 0.973564487885448 al eliminar el ítem Item 2 _ 8"
## [1] "El índice de confiabilidad cambia a 0.973834425944784 al eliminar el ítem Item 2 _ 9"
## [1] "El índice de confiabilidad cambia a 0.97473972477145 al eliminar el ítem Item 2 _ 10"
```

```
# Amacenas los valores
```

```
alfa = rbind(alfa,
             c("Flexibilidad - Pre",
```

```

        alpha(flexibilidad_pre[,20:32]))
alfa = rbind(alfa,
             c("Flexibilidad - Post",
               alpha(flexibilidad_post[,17:26]))))

```

3.3.4.2 Indicadores psicométricos

```

analitem = item.exam(flexibilidad_pre[,20:32], 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
0.442	0.682	0.621	0.734	0.643	0.301
0.394	0.695	0.644	0.808	0.530	0.274
0.442	0.804	0.763	0.734	0.726	0.355
0.445	0.826	0.789	0.728	0.762	0.368
0.391	0.735	0.689	0.811	0.545	0.287
0.423	0.832	0.798	0.767	0.677	0.352
0.432	0.839	0.806	0.752	0.716	0.362
0.441	0.843	0.810	0.736	0.744	0.372
0.363	0.745	0.704	0.844	0.455	0.270
0.392	0.743	0.698	0.811	0.543	0.291
0.443	0.838	0.803	0.731	0.755	0.371
0.438	0.823	0.786	0.742	0.718	0.360

0.444	0.819	0.780	0.730	0.731	0.364
-------	-------	-------	-------	-------	-------

```
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
0.495	0.904	0.880	0.571	0.996	0.447
0.484	0.898	0.874	0.624	0.977	0.435
0.492	0.924	0.905	0.588	1.000	0.455
0.494	0.915	0.894	0.577	0.989	0.452
0.491	0.889	0.862	0.597	0.985	0.436
0.495	0.918	0.897	0.572	0.998	0.454
0.495	0.918	0.898	0.570	0.992	0.455
0.488	0.918	0.898	0.610	0.996	0.448
0.495	0.911	0.889	0.572	0.996	0.451
0.498	0.890	0.863	0.549	0.998	0.443

```
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 = 0.287, 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 = 0.289, 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	1174	0.76379	0.33272	1.0	0.82308	0.00000	0	1	1
Total_post	1174	0.63910	0.43173	0.9	0.67372	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
  )

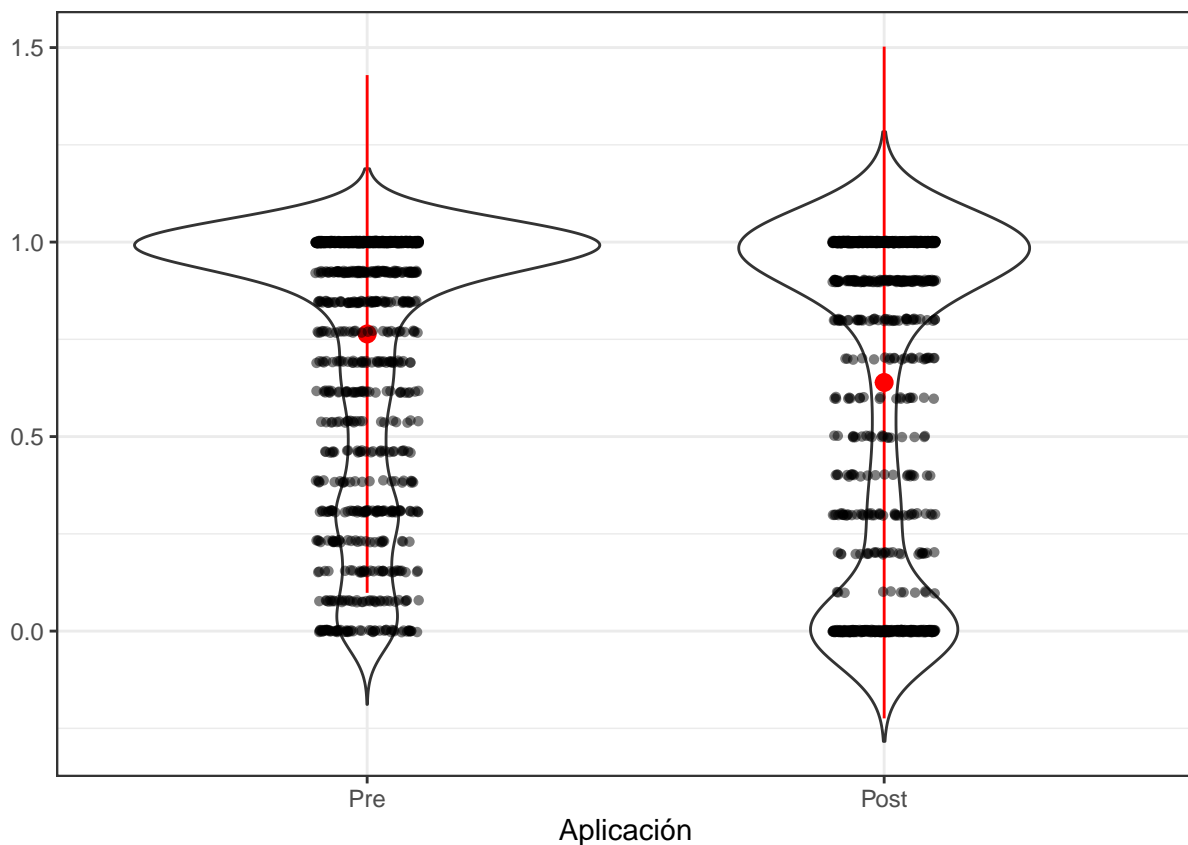
comparacion
```

3.3.4.3.3 Comparación de medias

```
##
```

```
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Total_pre and pre_post$Total_post
## V = 216454, 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/primer_aaplicacion/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: -0.32122 (small)
## 95 percent confidence interval:
##   lower   upper
## -0.39045 -0.25200

# 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[,1:17]

col_items = paste(rep("Situación", 11), c(rep(1, 3),
      rep(2:3, each = 4)), paste("Item", 1:11))
```

```

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

colnames(socioemocional_pre) = vector

socioemocional_pre = filter(socioemocional_pre,
                             !is.na(`Código`),
                             !is.na(`Situación 1 Item 1`),
                             `Situación 1 Item 1` != 'NULL')

socioemocional_pre[,7:17] = apply(socioemocional_pre[,7:17], 2,
                                  function(x) str_to_upper(x))

#Post

socioemocional_post = read_sheet(url_post, sheet = "Socioemocional")

socioemocional_post = socioemocional_post[,1:16]

col_items = socioemocional_post[1,7:16]

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

colnames(socioemocional_post) = vector

socioemocional_post = filter(socioemocional_post,
                             !is.na(`Código`),
                             !is.na(`1(flor)`),
#                             !is.na(`Nombre del estudiante`),
                             `1(flor)` != 'NULL')

socioemocional_post[,7:16] = apply(socioemocional_post[,7:16], 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`,
                `Situación 1 Item 1`,
                `Situación 2 Item 4`,
                `Situación 2 Item 6`,

```



```

        `Situación 3 Item 8`,
        `Situación 3 Item 11`)

claves_reconocimiento_pre = c("C", "A", "C", "A", "A")

socio_reconocimiento_pre[,2:6] = calificacion(data = socio_reconocimiento_pre[,2:6],
                                              claves = claves_reconocimiento_pre)

socio_reconocimiento_pre[,2:6] = as.data.frame(apply(socio_reconocimiento_pre[,2:6],
                                                    2, function(x) as.numeric(x)))

# Post

socio_reconocimiento_post = as_tibble(socioemocional_post) %>%
  dplyr::select(`Código`,
               `2(estrella)`,
               `5(pelota)`,
               `7(flecha)`,
               `8(cruz)`)

claves_reconocimiento_post = c("D", "A", "C", "B")

socio_reconocimiento_post[,2:5] = calificacion(data = socio_reconocimiento_post[,2:5],
                                              claves = claves_reconocimiento_post)

socio_reconocimiento_post[,2:5] = as.data.frame(apply(socio_reconocimiento_post[,2:5], 2, func

#####
# Calificación Regulación

# pre

regulacion_pre = as_tibble(socioemocional_pre) %>%
  dplyr::select(`Código`,
               `Situación 1 Item 2`,
               `Situación 2 Item 7`,
               `Situación 3 Item 9`)

regulacion_pre[2:4] = as.data.frame(apply(regulacion_pre[2:4], 2, function(x)
                                          str_replace_all(x, c("A" = "1", "B" = "1",
                                                                "C" = "0", "D" = "0",
                                                                "X" = "", "0" = ""))))

```

```

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(flor)`,
               `4(pulgar)`,
               `10(arbol)`)

regulacion_post$`1(flor)` = str_replace_all(regulacion_post$`1(flor)`,
                                             c("A" = "0", "B" = "0",
                                                "C" = "1", "D" = "1",
                                                "X" = "", "0" = ""))

regulacion_post$`4(pulgar)` = str_replace_all(regulacion_post$`4(pulgar)`,
                                             c("A" = "1", "B" = "1",
                                                "C" = "0", "D" = "0",
                                                "X" = "", "0" = ""))

regulacion_post$`10(arbol)` = str_replace_all(regulacion_post$`10(arbol)`,
                                             c("A" = "1", "B" = "0",
                                                "C" = "0", "D" = "1",
                                                "X" = "", "0" = ""))

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`,
               `Situación 1 Item 3`,
               `Situación 2 Item 5`,
               `Situación 3 Item 10`)

expresion_pre$`Situación 1 Item 3` = str_replace_all(expresion_pre$`Situación 1 Item 3`,
                                             c("A" = "0", "B" = "1",
                                                "C" = "1", "D" = "0",
                                                "X" = "", "0" = ""))

```

```

expresion_pre$`Situación 2 Item 5` = str_replace_all(expresion_pre$`Situación 2 Item 5`,
                                                    c("A" = "1", "B" = "0",
                                                      "C" = "0", "D" = "1",
                                                      "X" = "", "0" = ""))

expresion_pre$`Situación 3 Item 10` = str_replace_all(expresion_pre$`Situación 3 Item 10`,
                                                    c("A" = "0", "B" = "0",
                                                      "C" = "1", "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`,
                `3(luna)`,
                `6(corazon)`,
                `9(bus)`)

expresion_post$`3(luna)` = str_replace_all(expresion_post$`3(luna)`,
                                                    c("A" = "0", "B" = "1",
                                                      "C" = "0", "D" = "1",
                                                      "X" = "", "0" = ""))

expresion_post$`6(corazon)` = str_replace_all(expresion_post$`6(corazon)`,
                                                    c("A" = "1", "B" = "0",
                                                      "C" = "0", "D" = "1",
                                                      "X" = "", "0" = ""))

expresion_post$`9(bus)` = str_replace_all(expresion_post$`9(bus)`,
                                                    c("A" = "0", "B" = "1",
                                                      "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] 0.46317
```

```

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 0.431945307743447 al eliminar el ítem Situación 1"
## [1] "El índice de confiabilidad cambia a 0.463967322703798 al eliminar el ítem Situación 2"
## [1] "El índice de confiabilidad cambia a 0.435853739349457 al eliminar el ítem Situación 3"
## [1] "El índice de confiabilidad cambia a 0.483838279815517 al eliminar el ítem Situación 1"
## [1] "El índice de confiabilidad cambia a 0.412125212252934 al eliminar el ítem Situación 2"
## [1] "El índice de confiabilidad cambia a 0.424114407224096 al eliminar el ítem Situación 2"
## [1] "El índice de confiabilidad cambia a 0.434589904291102 al eliminar el ítem Situación 3"
## [1] "El índice de confiabilidad cambia a 0.414502130706953 al eliminar el ítem Situación 3"
## [1] "El índice de confiabilidad cambia a 0.462681233284309 al eliminar el ítem Situación 1"
## [1] "El índice de confiabilidad cambia a 0.40738695212266 al eliminar el ítem Situación 2"
## [1] "El índice de confiabilidad cambia a 0.454535271177364 al eliminar el ítem Situación 3"

```

```

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] 0.61479
```

```

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 0.581995939594812 al eliminar el ítem 1(flor)"
## [1] "El índice de confiabilidad cambia a 0.663796034785691 al eliminar el ítem 4(pulgar)"
## [1] "El índice de confiabilidad cambia a 0.574464971274539 al eliminar el ítem 10(arbol)"

```

```
## [1] "El índice de confiabilidad cambia a 0.589987682937046 al eliminar el ítem 2(estrella)"
## [1] "El índice de confiabilidad cambia a 0.577901854029885 al eliminar el ítem 5(pelota)"
## [1] "El índice de confiabilidad cambia a 0.581314554208158 al eliminar el ítem 7(flecha)"
## [1] "El índice de confiabilidad cambia a 0.562707367548231 al eliminar el ítem 8(cruz)"
## [1] "El índice de confiabilidad cambia a 0.56952650154639 al eliminar el ítem 3(luna)"
## [1] "El índice de confiabilidad cambia a 0.578069871420914 al eliminar el ítem 6(corazon)"
## [1] "El índice de confiabilidad cambia a 0.603992246077752 al eliminar el ítem 9(bus)"
```

```
# 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] 0.242
```

```
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 0.233137519087398 al eliminar el ítem Situación 1"
## [1] "El índice de confiabilidad cambia a 0.203230956672435 al eliminar el ítem Situación 2"
## [1] "El índice de confiabilidad cambia a 0.0674591355049508 al eliminar el ítem Situación 3"
```

```
alpha(dplyr::select(regulacion_post, -`Código`))
```

3.4.1.4 Regulación post

```
## [1] 0.20543
```

```
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 -0.0352300382431334 al eliminar el ítem 1(flor)"
```

```
## [1] "El índice de confiabilidad cambia a 0.551163515720607 al eliminar el ítem 4(pulgar)"
## [1] "El índice de confiabilidad cambia a -0.196572837256662 al eliminar el ítem 10(arbol)"
```

```
alpha(dplyr::select(expresion_pre, -`Código`))
```

3.4.1.5 Expresión pre

```
## [1] 0.32742
```

```
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 0.325756023806977 al eliminar el ítem Situación 1"
## [1] "El índice de confiabilidad cambia a 0.288334559209835 al eliminar el ítem Situación 2"
## [1] "El índice de confiabilidad cambia a 0.139640272446718 al eliminar el ítem Situación 3"
```

```
alpha(dplyr::select(expresion_post, -`Código`))
```

3.4.1.6 Expresión post

```
## [1] 0.63893
```

```
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 0.510905243514338 al eliminar el ítem 3(luna)"
## [1] "El índice de confiabilidad cambia a 0.488035246350102 al eliminar el ítem 6(corazon)"
## [1] "El índice de confiabilidad cambia a 0.615860204622767 al eliminar el ítem 9(bus)"
```

```
alpha(dplyr::select(socio_reconocimiento_pre, -`Código`))
```

3.4.1.7 Reconocimiento pre

```
## [1] 0.48477
```

```
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 0.493493482284101 al eliminar el ítem Situación 1"
## [1] "El índice de confiabilidad cambia a 0.398446332309345 al eliminar el ítem Situación 2"
## [1] "El índice de confiabilidad cambia a 0.353107852797743 al eliminar el ítem Situación 2"
## [1] "El índice de confiabilidad cambia a 0.460329170624445 al eliminar el ítem Situación 3"
## [1] "El índice de confiabilidad cambia a 0.425510763617409 al eliminar el ítem Situación 3"
```

```
alpha(dplyr::select(socio_reconocimiento_post, -`Código`))
```

3.4.1.8 Reconocimiento post

```
## [1] 0.67493
```

```
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 0.60952588297544 al eliminar el ítem 2(estrella)"
## [1] "El índice de confiabilidad cambia a 0.627029770623648 al eliminar el ítem 5(pelota)"
## [1] "El índice de confiabilidad cambia a 0.611609156240214 al eliminar el ítem 7(flecha)"
## [1] "El índice de confiabilidad cambia a 0.588574354787975 al eliminar el ítem 8(cruz)"
```

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
0.382	0.381	0.201	0.823	0.315	0.145
0.499	0.360	0.118	0.462	0.394	0.180
0.460	0.402	0.185	0.698	0.388	0.185
0.461	0.263	0.035	0.305	0.260	0.121
0.500	0.483	0.257	0.521	0.550	0.241
0.470	0.443	0.226	0.672	0.483	0.208
0.441	0.402	0.195	0.736	0.373	0.177
0.489	0.474	0.252	0.396	0.520	0.232
0.420	0.299	0.093	0.772	0.272	0.125
0.463	0.481	0.275	0.690	0.486	0.223
0.459	0.361	0.140	0.699	0.370	0.165

```
# 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
0.353	0.484	0.319	0.854	0.317	0.171
0.451	0.232	-0.010	0.285	0.220	0.105
0.434	0.547	0.352	0.749	0.531	0.237
0.409	0.479	0.283	0.788	0.404	0.196
0.470	0.547	0.332	0.671	0.587	0.257
0.462	0.533	0.319	0.693	0.599	0.246
0.408	0.571	0.394	0.789	0.497	0.233
0.313	0.532	0.395	0.890	0.298	0.166
0.335	0.496	0.343	0.872	0.304	0.166
0.247	0.360	0.237	0.935	0.155	0.089

```
# 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 = 0.228, 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 = 0.312, 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	875	0.63200	0.30245	0.66667	0.65478	0.4942	0	1	1
Total_post	875	0.60724	0.26924	0.66667	0.62482	0.0000	0	1	1

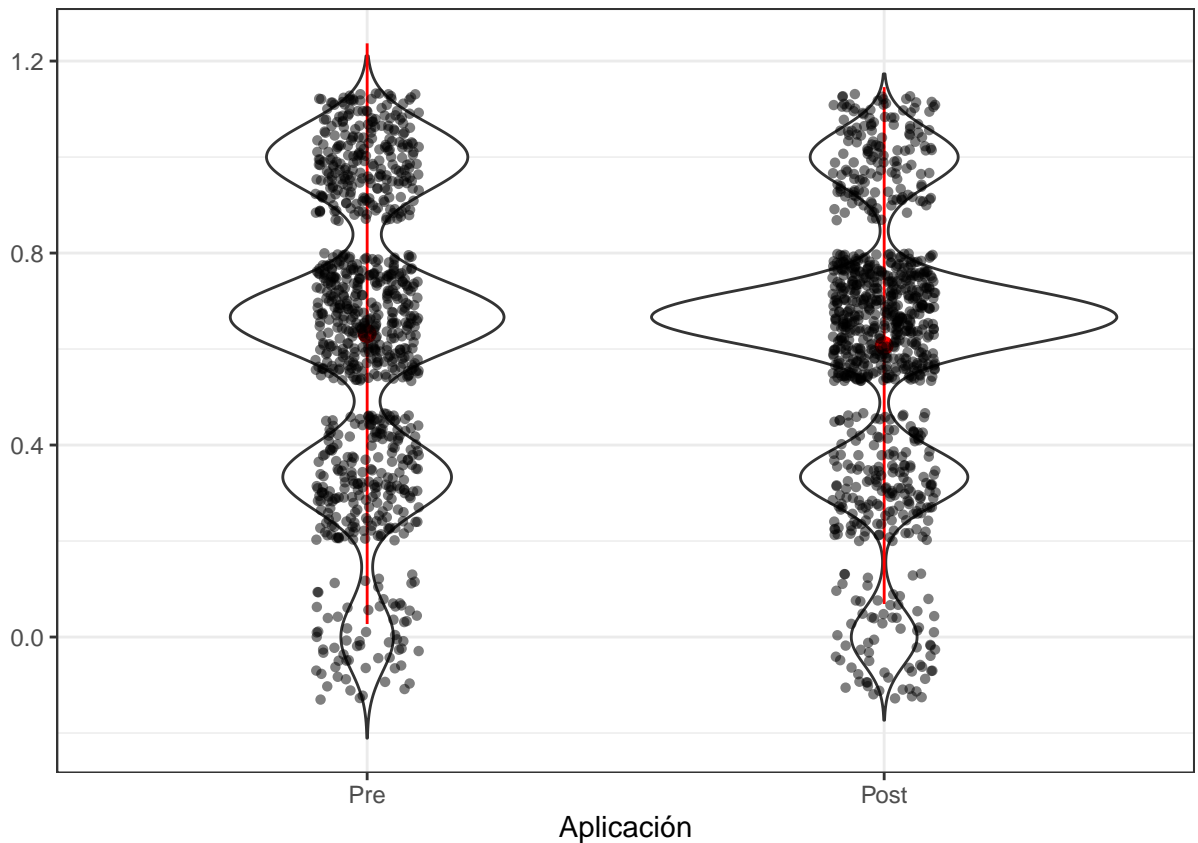
```
comparacion =
  wilcox.test(
    x      = pre_post$Total_pre,
    y      = pre_post$Total_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )
```

```
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 = 95506, p-value = 0.0016
## 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/primer_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: -0.086442 (negligible)
## 95 percent confidence interval:
##      lower      upper
## -0.1744194  0.0015361
```

```
# 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[,2:6], 1,
  function(x) sum(x, na.rm= FALSE))

socio_reconocimiento_post$Total_post = apply(socio_reconocimiento_post[,2:5], 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 = 0.152, 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 = 0.241, 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	875	0.49874	0.27235	0.60	0.50499	0.29652	0	1	1
Total_post	875	0.71743	0.31676	0.75	0.76462	0.37065	0	1	1

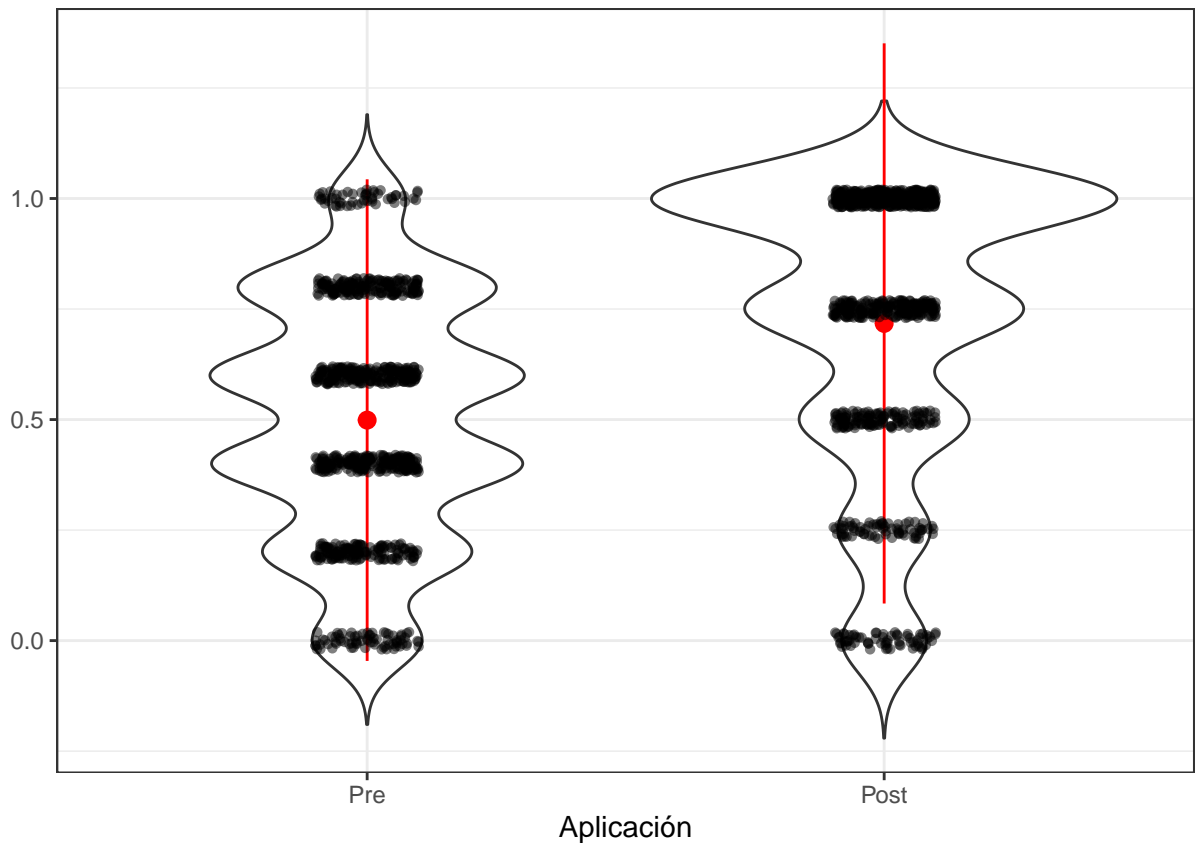
```
comparacion =
  wilcox.test(
    x      = pre_post$Total_pre,
    y      = pre_post$Total_post,
    alternative = "two.sided",
    mu      = 0,
    var.equal = TRUE,
    paired   = TRUE,
    conf.level = 0.95
  )

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 = 63838, 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/primer_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: 0.73898 (medium)
## 95 percent confidence interval:
##   lower   upper
## 0.64696 0.83100
```

```
# 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 = 0.226, 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 = 0.459, 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	875	0.67657	0.31635	0.66667	0.71517	0.4942	0	1	1
Total_post	875	0.88038	0.25446	1.00000	0.94912	0.0000	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
```

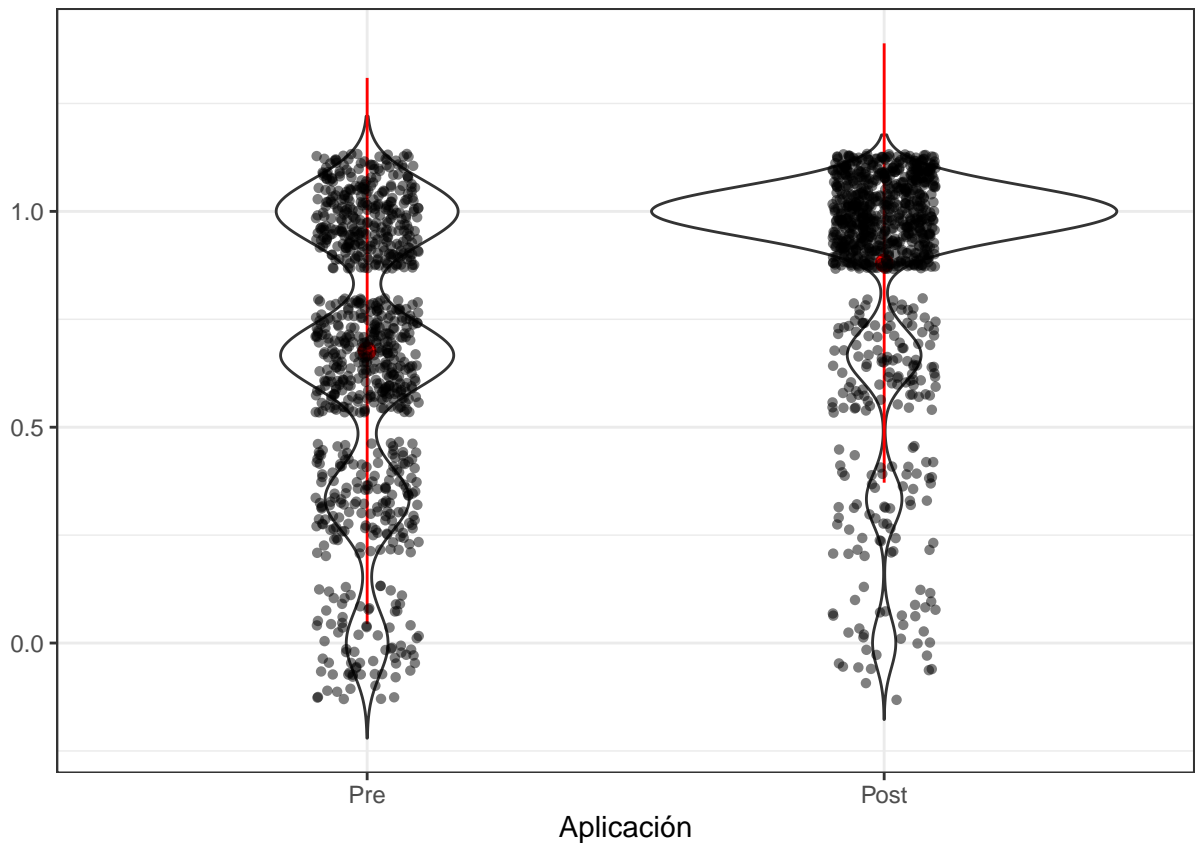
3.4.3.3.3 Comparación de medias

```

##
## Wilcoxon signed rank test with continuity correction
##
## data: pre_post$Total_pre and pre_post$Total_post
## V = 13439, 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/primer_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: 0.70663 (medium)
## 95 percent confidence interval:
##   lower   upper
## 0.61818 0.79508
```

```
# 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	0.80
Actitudes lenguaje post	0.80
Actitudes matematicas pre	0.87
Actitudes matematicas post	0.83
Motivacion pre	0.79
Motivacion post	0.62
Memoria auditiva (Item 1) - Pre	0.71
Memoria auditiva (Item 2) - Pre	0.69
Memoria auditiva (Item 1) - Post	0.68
Memoria auditiva (Item 2) - Post	0.56
Memoria visual (Item 1) - Pre	0.44
Memoria visual (Item 2) - Pre	0.54
Memoria visual (Item 1) - Post	0.47
Memoria visual (Item 2) - Post	0.58
Inhibicion - Pre	0.95

Inhibicion - Post	0.93
Flexibilidad - Pre	0.95
Flexibilidad - Post	0.98
Socioemocionales - Pre	0.46
Socioemocionales - Post	0.61

```
# Exporte
```

```
write.xlsx(drop_na(alfa), "../Data/processed/primer_a_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
434	0.77	0.26	0	1	Actitudes hacia el lenguaje - Total	Pre
434	0.82	0.23	0	1	Actitudes hacia el lenguaje - Total	Post
434	0.74	0.31	0	1	Actitudes hacia el lenguaje - Afectivo	Pre
434	0.83	0.26	0	1	Actitudes hacia el lenguaje - Afectivo	Post
434	0.81	0.28	0	1	Actitudes hacia el lenguaje - Cognitivo	Pre
434	0.83	0.26	0	1	Actitudes hacia el lenguaje - Cognitivo	Post
434	0.77	0.30	0	1	Actitudes hacia el lenguaje - Conativo	Pre
434	0.81	0.27	0	1	Actitudes hacia el lenguaje - Conativo	Post
468	0.85	0.25	0	1	Actitudes hacia las matematicas - Total	Pre
468	0.87	0.22	0	1	Actitudes hacia las matematicas - Total	Post
468	0.83	0.29	0	1	Actitudes hacia las matematicas - Afectivo	Pre

468	0.85	0.25	0	1	Actitudes hacia las matematicas - Afectivo	Post
468	0.87	0.26	0	1	Actitudes hacia las matematicas - Cognitivo	Pre
468	0.89	0.24	0	1	Actitudes hacia las matematicas - Cognitivo	Post
468	0.84	0.29	0	1	Actitudes hacia las matematicas - Conativo	Pre
468	0.87	0.25	0	1	Actitudes hacia las matematicas - Conativo	Post
882	0.84	0.20	0	1	Motivacion - Total	Pre
882	0.78	0.19	0	1	Motivacion - Total	Post
882	0.90	0.21	0	1	Motivacion - Interes	Pre
882	0.87	0.25	0	1	Motivacion - Interes	Post
882	0.22	0.27	0	1	Motivacion - Metas	Pre
882	0.29	0.29	0	1	Motivacion - Metas	Post
882	0.20	0.32	0	1	Motivacion - Atribucion externa	Pre
882	0.31	0.34	0	1	Motivacion - Atribucion externa	Post
882	0.87	0.27	0	1	Motivacion - Expectativas	Pre
882	0.84	0.31	0	1	Motivacion - Expectativas	Post
906	0.64	0.24	0	1	Memoria auditiva	Pre
906	0.68	0.20	0	1	Memoria auditiva	Post
904	0.38	0.29	0	1	Memoria visual	Pre
904	0.61	0.30	0	1	Memoria visual	Post
905	0.78	0.32	0	1	Inhibicion	Pre
905	0.87	0.26	0	1	Inhibicion	Post
1174	0.76	0.33	0	1	Flexibilidad	Pre
1174	0.64	0.43	0	1	Flexibilidad	Post
875	0.63	0.30	0	1	Socioemocionales - Regulacion	Pre
875	0.61	0.27	0	1	Socioemocionales - Regulacion	Post
875	0.50	0.27	0	1	Socioemocionales - Reconocimiento	Pre
875	0.72	0.32	0	1	Socioemocionales - Reconocimiento	Post
875	0.68	0.32	0	1	Socioemocionales - Expresion	Pre
875	0.88	0.25	0	1	Socioemocionales - Expresion	Post

```
# Exporte
```

```
write.xlsx(descriptivos, "../Data/processed/primer_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
0.73	0.54	Actitudes hacia el lenguaje - Pre	Grupo 1 __ 1
0.86	0.37	Actitudes hacia el lenguaje - Pre	Grupo 1 __ 2
0.79	0.52	Actitudes hacia el lenguaje - Pre	Grupo 1 __ 3
0.80	0.46	Actitudes hacia el lenguaje - Pre	Grupo 2 __ 1
0.88	0.30	Actitudes hacia el lenguaje - Pre	Grupo 2 __ 2
0.82	0.45	Actitudes hacia el lenguaje - Pre	Grupo 2 __ 3
0.66	0.72	Actitudes hacia el lenguaje - Pre	Grupo 3 __ 1
0.67	0.76	Actitudes hacia el lenguaje - Pre	Grupo 3 __ 2
0.70	0.70	Actitudes hacia el lenguaje - Pre	Grupo 3 __ 3
0.92	0.22	Actitudes hacia el lenguaje - Post	Grupo 1 __ 1
0.87	0.33	Actitudes hacia el lenguaje - Post	Grupo 1 __ 2
0.82	0.39	Actitudes hacia el lenguaje - Post	Grupo 1 __ 3
0.71	0.73	Actitudes hacia el lenguaje - Post	Grupo 2 __ 1
0.71	0.73	Actitudes hacia el lenguaje - Post	Grupo 2 __ 2
0.74	0.65	Actitudes hacia el lenguaje - Post	Grupo 2 __ 3
0.85	0.40	Actitudes hacia el lenguaje - Post	Grupo 3 __ 1
0.89	0.31	Actitudes hacia el lenguaje - Post	Grupo 3 __ 2
0.86	0.37	Actitudes hacia el lenguaje - Post	Grupo 3 __ 3
0.78	0.54	Actitudes hacia las matematicas - Pre	Grupo 1 __ 1
0.88	0.37	Actitudes hacia las matematicas - Pre	Grupo 1 __ 2
0.81	0.52	Actitudes hacia las matematicas - Pre	Grupo 1 __ 3
0.89	0.32	Actitudes hacia las matematicas - Pre	Grupo 2 __ 1
0.87	0.36	Actitudes hacia las matematicas - Pre	Grupo 2 __ 2
0.87	0.36	Actitudes hacia las matematicas - Pre	Grupo 2 __ 3
0.81	0.51	Actitudes hacia las matematicas - Pre	Grupo 3 __ 1
0.85	0.44	Actitudes hacia las matematicas - Pre	Grupo 3 __ 2

0.83	0.50	Actitudes hacia las matematicas - Pre	Grupo 3 _ 3
0.80	0.51	Actitudes hacia las matematicas - Post	Grupo 1 _ 1
0.90	0.26	Actitudes hacia las matematicas - Post	Grupo 1 _ 2
0.82	0.48	Actitudes hacia las matematicas - Post	Grupo 1 _ 3
0.87	0.37	Actitudes hacia las matematicas - Post	Grupo 2 _ 1
0.89	0.31	Actitudes hacia las matematicas - Post	Grupo 2 _ 2
0.87	0.36	Actitudes hacia las matematicas - Post	Grupo 2 _ 3
0.86	0.39	Actitudes hacia las matematicas - Post	Grupo 3 _ 1
0.86	0.39	Actitudes hacia las matematicas - Post	Grupo 3 _ 2
0.89	0.34	Actitudes hacia las matematicas - Post	Grupo 3 _ 3
0.86	0.32	Motivacion - Pre	1
0.69	0.60	Motivacion - Pre	2
0.87	0.33	Motivacion - Pre	3
0.92	0.22	Motivacion - Pre	4
0.89	0.30	Motivacion - Pre	5
0.88	0.31	Motivacion - Pre	6
0.91	0.22	Motivacion - Pre	7
0.81	0.46	Motivacion - Pre	8
0.82	0.43	Motivacion - Pre	9
0.89	0.31	Motivacion - Pre	10
0.72	0.59	Motivacion - Pre	11
0.77	0.54	Motivacion - Pre	12
0.84	0.35	Motivacion - Post	1
0.76	0.49	Motivacion - Post	2
0.69	0.49	Motivacion - Post	3
0.89	0.29	Motivacion - Post	4
0.75	0.44	Motivacion - Post	5
0.84	0.31	Motivacion - Post	6
0.88	0.30	Motivacion - Post	7
0.83	0.40	Motivacion - Post	8
0.59	0.53	Motivacion - Post	9
0.66	0.38	Motivacion - Post	10
0.84	0.39	Memoria auditiva (Item 1) - Pre	Caballo
0.66	0.58	Memoria auditiva (Item 1) - Pre	Perro
0.67	0.62	Memoria auditiva (Item 1) - Pre	Aguila
0.79	0.47	Memoria auditiva (Item 1) - Pre	Pollito
0.60	0.64	Memoria auditiva (Item 1) - Pre	Foca
0.64	0.55	Memoria auditiva (Item 1) - Pre	Vaca
0.72	0.54	Memoria auditiva (Item 1) - Pre	Cocodrilo
0.61	0.59	Memoria auditiva (Item 1) - Pre	Sapo
0.68	0.52	Memoria auditiva (Item 2) - Pre	Oveja
0.61	0.64	Memoria auditiva (Item 2) - Pre	Elefante
0.75	0.48	Memoria auditiva (Item 2) - Pre	Tiburon
0.69	0.57	Memoria auditiva (Item 2) - Pre	Caracol
0.70	0.59	Memoria auditiva (Item 2) - Pre	Raton
0.50	0.64	Memoria auditiva (Item 2) - Pre	Gato
0.70	0.56	Memoria auditiva (Item 2) - Pre	Tortuga
0.69	0.52	Memoria auditiva (Item 2) - Pre	Pez

0.82	0.36	Memoria auditiva (Item 1) - Post	Caballo
0.80	0.42	Memoria auditiva (Item 1) - Post	Gallina
0.80	0.40	Memoria auditiva (Item 1) - Post	Conejo
0.74	0.52	Memoria auditiva (Item 1) - Post	Leopardo
0.74	0.55	Memoria auditiva (Item 1) - Post	Ardilla
0.68	0.58	Memoria auditiva (Item 1) - Post	Cerdo
0.75	0.53	Memoria auditiva (Item 1) - Post	Cangrejo
0.64	0.53	Memoria auditiva (Item 1) - Post	Burro
0.91	0.21	Memoria auditiva (Item 2) - Post	Estrella
0.63	0.58	Memoria auditiva (Item 2) - Post	Gato
0.57	0.58	Memoria auditiva (Item 2) - Post	Abeja
0.54	0.68	Memoria auditiva (Item 2) - Post	Cebra
0.68	0.49	Memoria auditiva (Item 2) - Post	Rana
0.65	0.54	Memoria auditiva (Item 2) - Post	Armadillo
0.74	0.44	Memoria auditiva (Item 2) - Post	Tortuga
0.78	0.37	Memoria auditiva (Item 2) - Post	Iguana
0.44	0.75	Memoria visual (Item 1) - Pre	Gym
0.43	0.78	Memoria visual (Item 1) - Pre	Museo
0.63	0.64	Memoria visual (Item 1) - Pre	Cajas
0.42	0.76	Memoria visual (Item 2) - Pre	Salon
0.55	0.84	Memoria visual (Item 2) - Pre	Aeropuerto
0.48	0.76	Memoria visual (Item 2) - Pre	Iglesia
0.44	0.89	Memoria visual (Item 1) - Post	Iglesia
0.81	0.52	Memoria visual (Item 1) - Post	GYM
0.73	0.72	Memoria visual (Item 1) - Post	Zoo
0.82	0.47	Memoria visual (Item 2) - Post	Vet
0.78	0.58	Memoria visual (Item 2) - Post	Playa
0.61	0.87	Memoria visual (Item 2) - Post	Fruteria
0.77	0.61	Inhibicion - Pre	Item 2 __ 1
0.78	0.60	Inhibicion - Pre	Item 2 __ 2
0.77	0.64	Inhibicion - Pre	Item 2 __ 3
0.79	0.61	Inhibicion - Pre	Item 2 __ 4
0.75	0.67	Inhibicion - Pre	Item 2 __ 5
0.75	0.64	Inhibicion - Pre	Item 2 __ 6
0.75	0.64	Inhibicion - Pre	Item 2 __ 7
0.77	0.65	Inhibicion - Pre	Item 2 __ 8
0.78	0.61	Inhibicion - Pre	Item 2 __ 9
0.77	0.63	Inhibicion - Pre	Item 2 __ 10
0.76	0.67	Inhibicion - Pre	Item 2 __ 11
0.76	0.64	Inhibicion - Pre	Item 2 __ 12
0.75	0.67	Inhibicion - Pre	Item 2 __ 13
0.90	0.29	Inhibicion - Post	Item 2 __ 1
0.85	0.45	Inhibicion - Post	Item 2 __ 2
0.84	0.47	Inhibicion - Post	Item 2 __ 3
0.85	0.44	Inhibicion - Post	Item 2 __ 4
0.85	0.45	Inhibicion - Post	Item 2 __ 5
0.85	0.46	Inhibicion - Post	Item 2 __ 6
0.86	0.42	Inhibicion - Post	Item 2 __ 7

0.86	0.42	Inhibicion - Post	Item 2 _ 8
0.86	0.41	Inhibicion - Post	Item 2 _ 9
0.84	0.48	Inhibicion - Post	Item 2 _ 10
0.73	0.64	Flexibilidad - Pre	Item 2 _ 1
0.81	0.53	Flexibilidad - Pre	Item 2 _ 2
0.73	0.73	Flexibilidad - Pre	Item 2 _ 3
0.73	0.76	Flexibilidad - Pre	Item 2 _ 4
0.81	0.55	Flexibilidad - Pre	Item 2 _ 5
0.77	0.68	Flexibilidad - Pre	Item 2 _ 6
0.75	0.72	Flexibilidad - Pre	Item 2 _ 7
0.74	0.74	Flexibilidad - Pre	Item 2 _ 8
0.84	0.45	Flexibilidad - Pre	Item 2 _ 9
0.81	0.54	Flexibilidad - Pre	Item 2 _ 10
0.73	0.75	Flexibilidad - Pre	Item 2 _ 11
0.74	0.72	Flexibilidad - Pre	Item 2 _ 12
0.73	0.73	Flexibilidad - Pre	Item 2 _ 13
0.57	1.00	Flexibilidad - Post	Item 2 _ 1
0.62	0.98	Flexibilidad - Post	Item 2 _ 2
0.59	1.00	Flexibilidad - Post	Item 2 _ 3
0.58	0.99	Flexibilidad - Post	Item 2 _ 4
0.60	0.99	Flexibilidad - Post	Item 2 _ 5
0.57	1.00	Flexibilidad - Post	Item 2 _ 6
0.57	0.99	Flexibilidad - Post	Item 2 _ 7
0.61	1.00	Flexibilidad - Post	Item 2 _ 8
0.57	1.00	Flexibilidad - Post	Item 2 _ 9
0.55	1.00	Flexibilidad - Post	Item 2 _ 10
0.82	0.31	Socioemocionales - Pre	Situacion 1 Item 2
0.46	0.39	Socioemocionales - Pre	Situacion 2 Item 7
0.70	0.39	Socioemocionales - Pre	Situacion 3 Item 9
0.31	0.26	Socioemocionales - Pre	Situacion 1 Item 1
0.52	0.55	Socioemocionales - Pre	Situacion 2 Item 4
0.67	0.48	Socioemocionales - Pre	Situacion 2 Item 6
0.74	0.37	Socioemocionales - Pre	Situacion 3 Item 8
0.40	0.52	Socioemocionales - Pre	Situacion 3 Item 11
0.77	0.27	Socioemocionales - Pre	Situacion 1 Item 3
0.69	0.49	Socioemocionales - Pre	Situacion 2 Item 5
0.70	0.37	Socioemocionales - Pre	Situacion 3 Item 10
0.85	0.32	Socioemocionales - Post	1(flor)
0.28	0.22	Socioemocionales - Post	4(pulgar)
0.75	0.53	Socioemocionales - Post	10(arbol)
0.79	0.40	Socioemocionales - Post	2(estrella)
0.67	0.59	Socioemocionales - Post	5(pelota)
0.69	0.60	Socioemocionales - Post	7(flecha)
0.79	0.50	Socioemocionales - Post	8(cruz)
0.89	0.30	Socioemocionales - Post	3(luna)
0.87	0.30	Socioemocionales - Post	6(corazon)
0.93	0.16	Socioemocionales - Post	9(bus)

```
# Exporte

write.xlsx(indicadores_psicometricos,
           "../Data/processed/primer_a_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	0.77163	0.82309	0.00	0.207062
Actitudes hacia el lenguaje - Afectivo	0.73733	0.83180	0.00	0.328170
Actitudes hacia el lenguaje - Cognitivo	0.80722	0.82565	0.37	0.068602
Actitudes hacia el lenguaje - Conativo	0.77035	0.81183	0.01	0.142860
Actitudes hacia las matematicas - Total	0.84639	0.87132	0.02	0.105035
Actitudes hacia las matematicas - Afectivo	0.82550	0.85399	0.07	0.104724
Actitudes hacia las matematicas - Cognitivo	0.87322	0.89174	0.29	0.073709
Actitudes hacia las matematicas - Conativo	0.84046	0.86823	0.05	0.101462
Motivacion - Total	0.83796	0.77948	0.00	-0.297267
Motivacion - Interes	0.90108	0.87491	0.00	-0.111850
Motivacion - Metas	0.22137	0.29327	0.00	0.258145
Motivacion - Atribucion externa	0.20238	0.31179	0.00	0.333253

Motivacion - Expectativas	0.87075	0.83673	0.01	-0.116390
Memoria auditiva	0.64480	0.67777	0.00	0.148263
Memoria visual	0.37795	0.61486	0.00	0.806544
Inhibicion	0.78436	0.87094	0.00	0.298549
Flexibilidad	0.76379	0.63910	0.00	-0.321223
Socioemocionales - Regulacion	0.63200	0.60724	0.00	-0.086442
Socioemocionales - Reconocimiento	0.49874	0.71743	0.00	0.738978
Socioemocionales - Expresion	0.67657	0.88038	0.00	0.706628

```
# Exporte
```

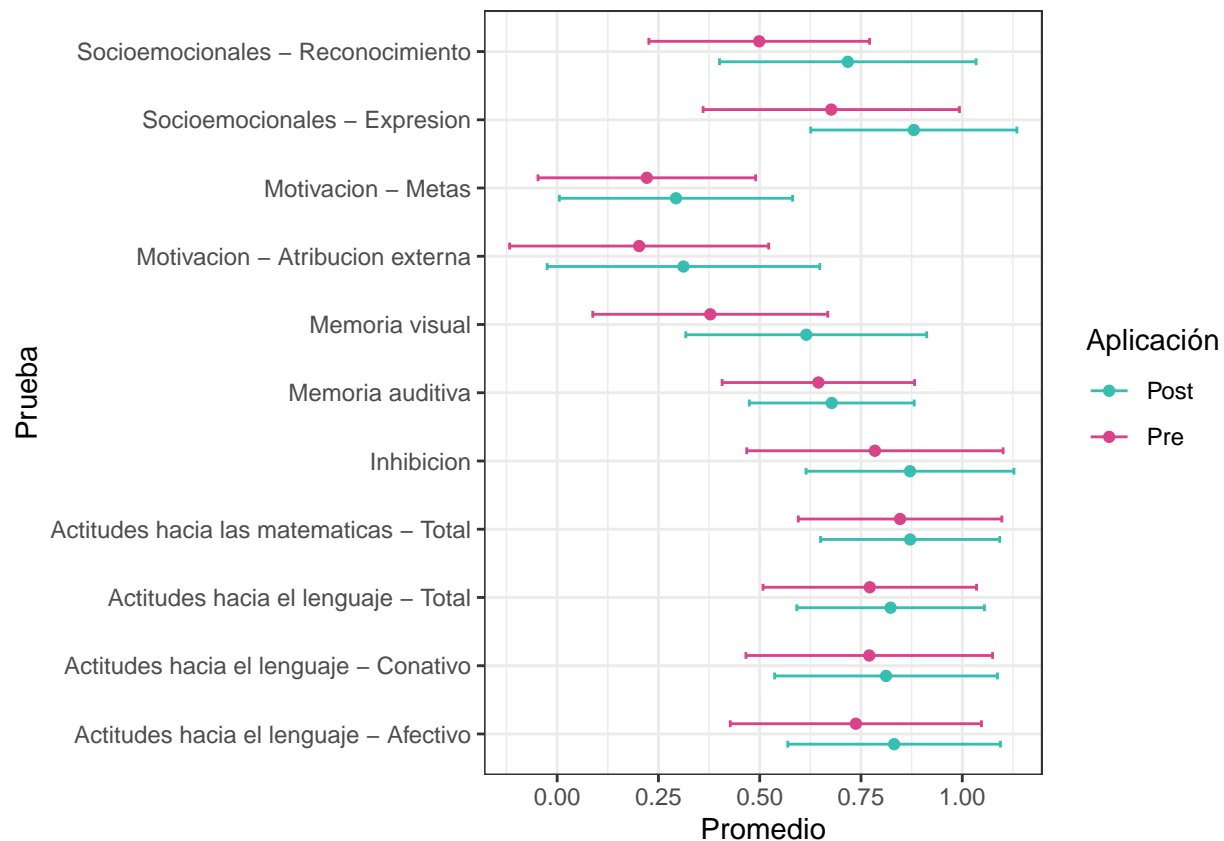
```
write.xlsx(comparaciones,
           "../Data/processed/primer_aaplicacion/pruebas_pre_post.xlsx",
           row.names = FALSE)
```

```
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()
```



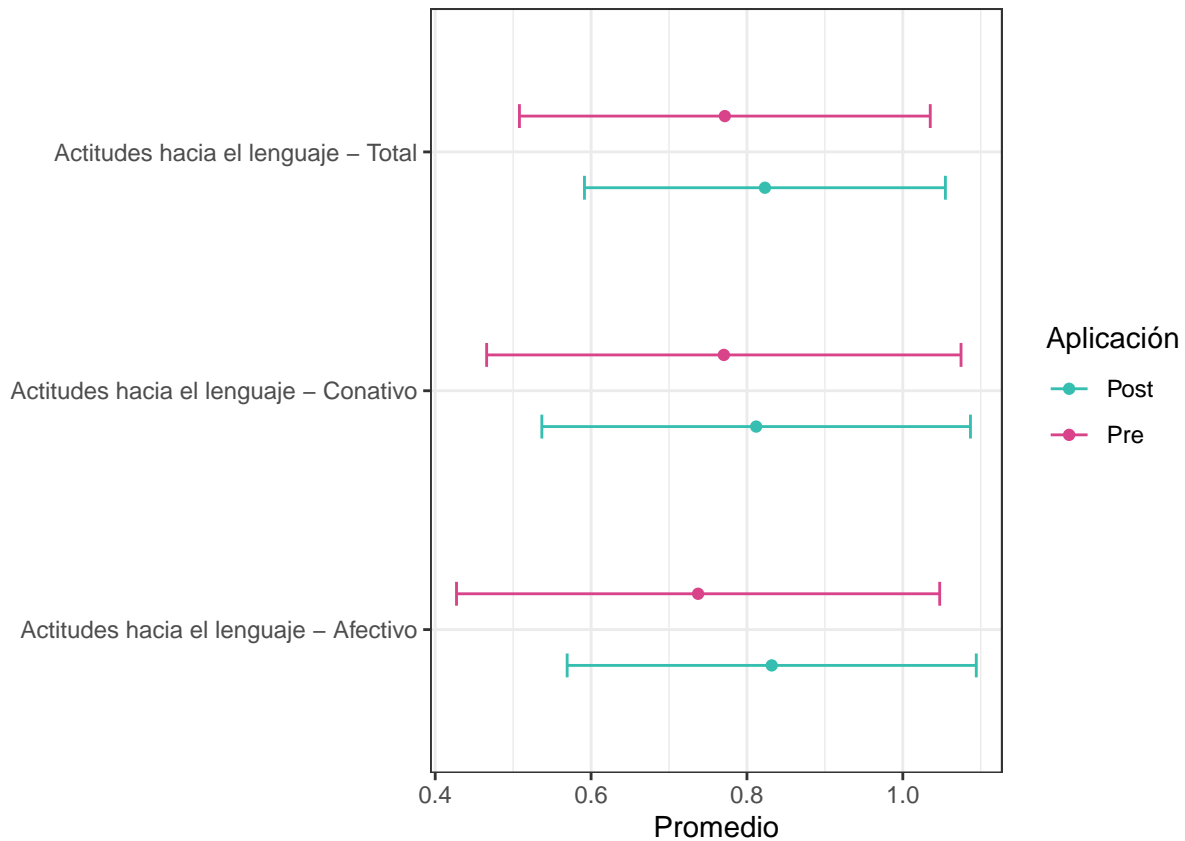
```
# Guardamos la imagen
```

```
ggsave("../Plots/primer_aaplicacion/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()
```



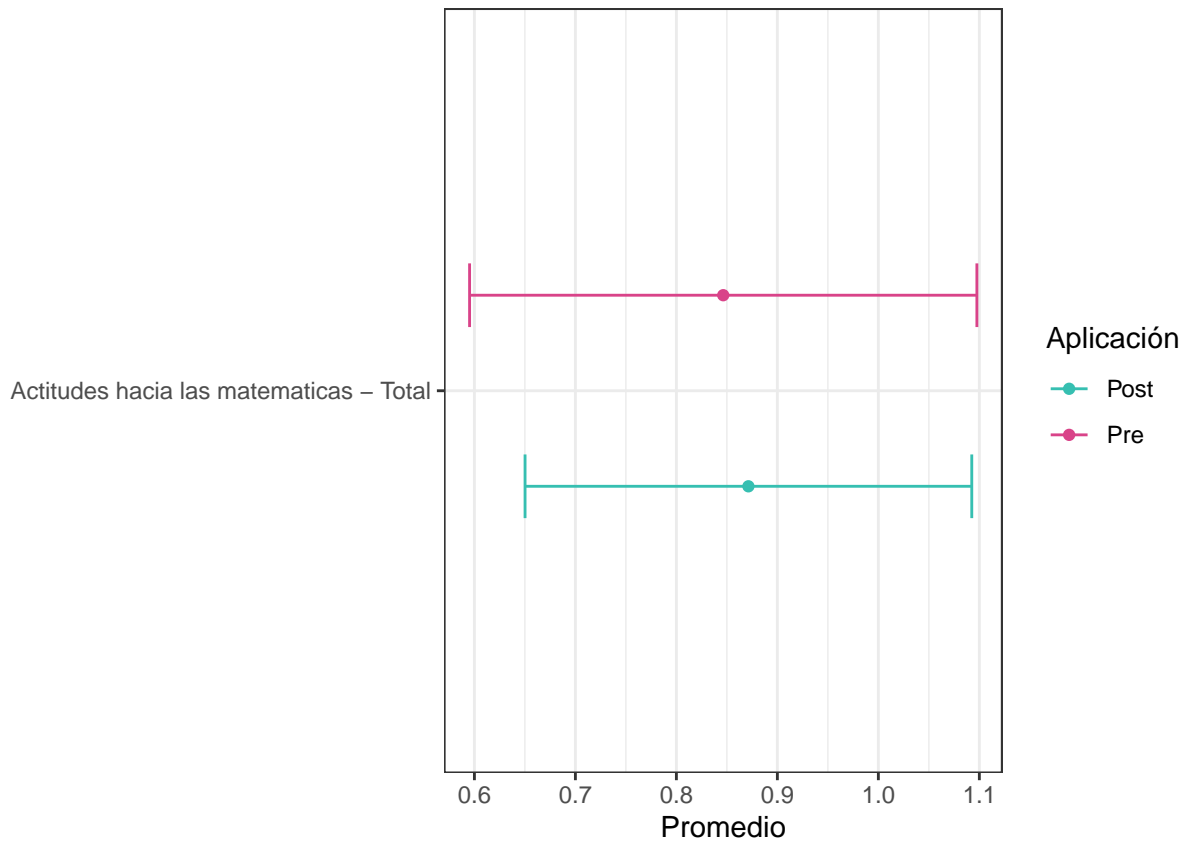
```
# Guardamos la imagen
```

```
ggsave("../Plots/primer_aaplicacion/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()
```

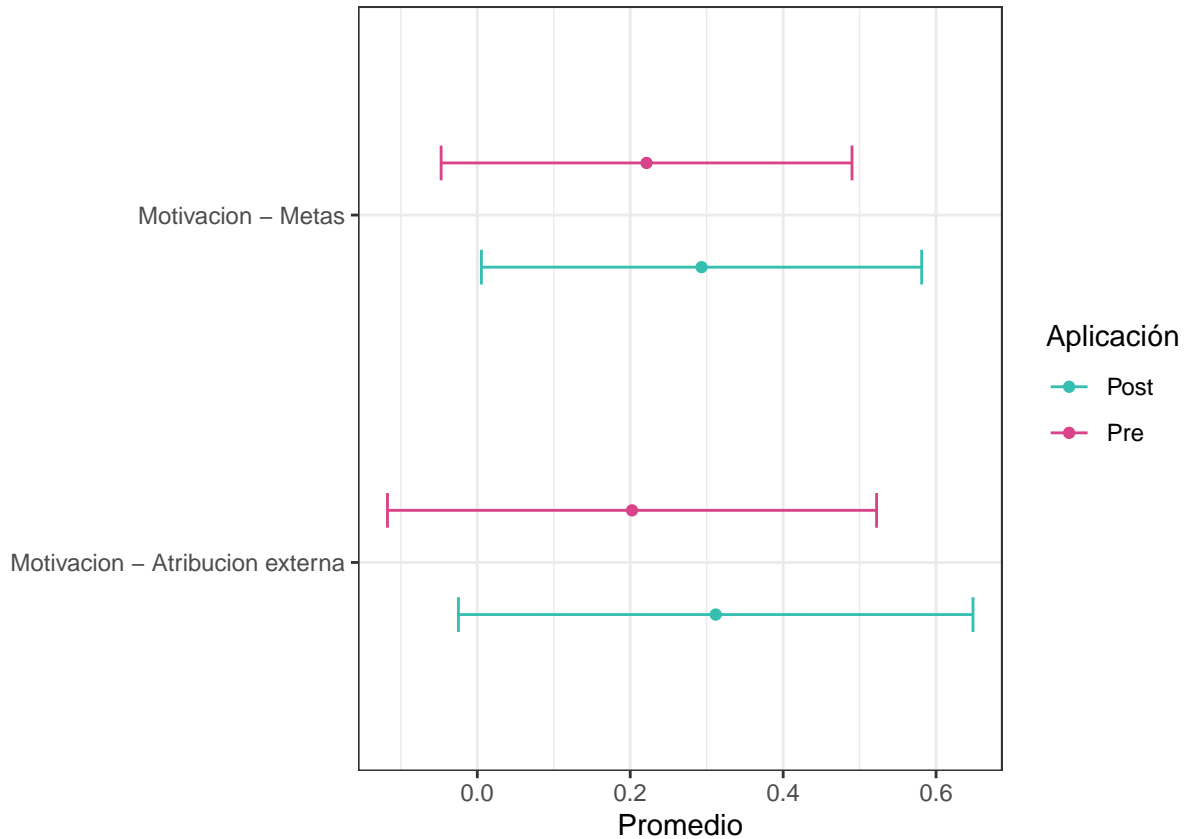
```
# Guardamos la imagen
```

```
ggsave("../Plots/primer_aaplicacion/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()
```



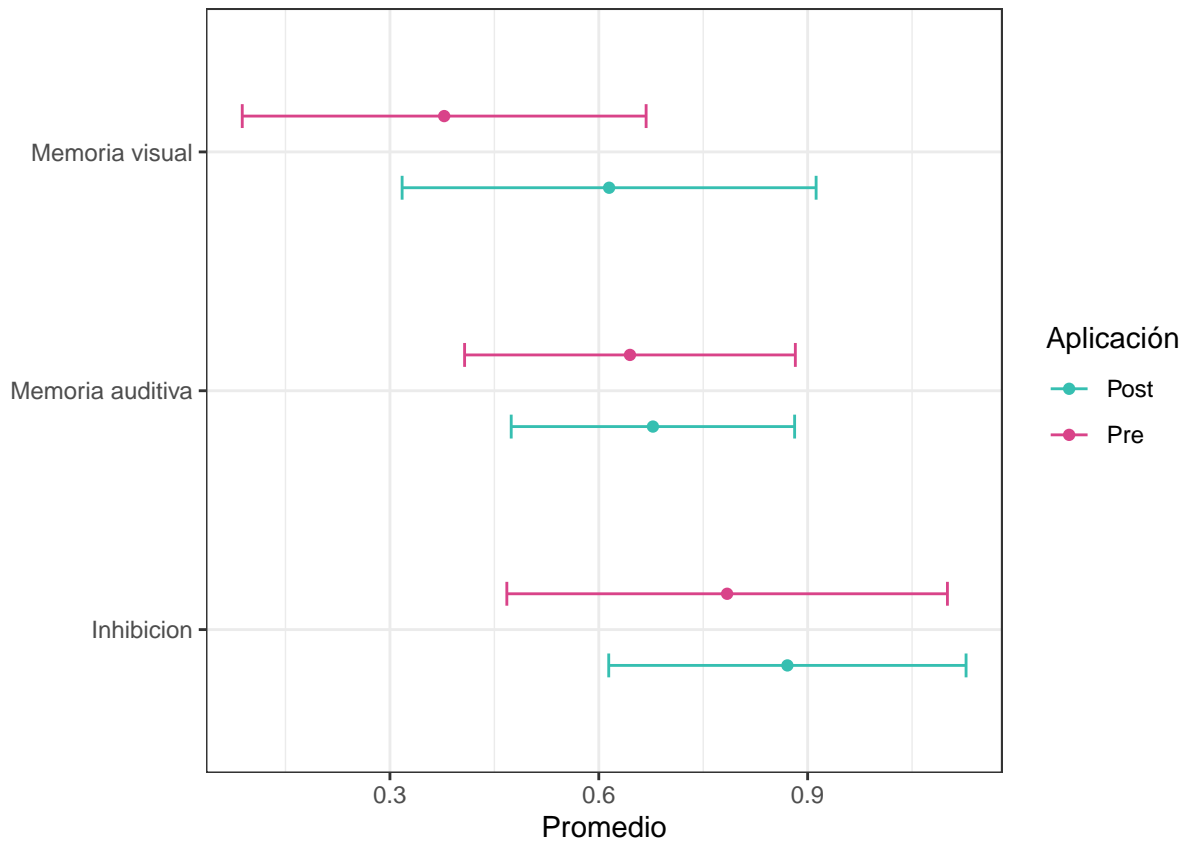
```
# Guardamos la imagen
```

```
ggsave("../Plots/primer_a 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()
```



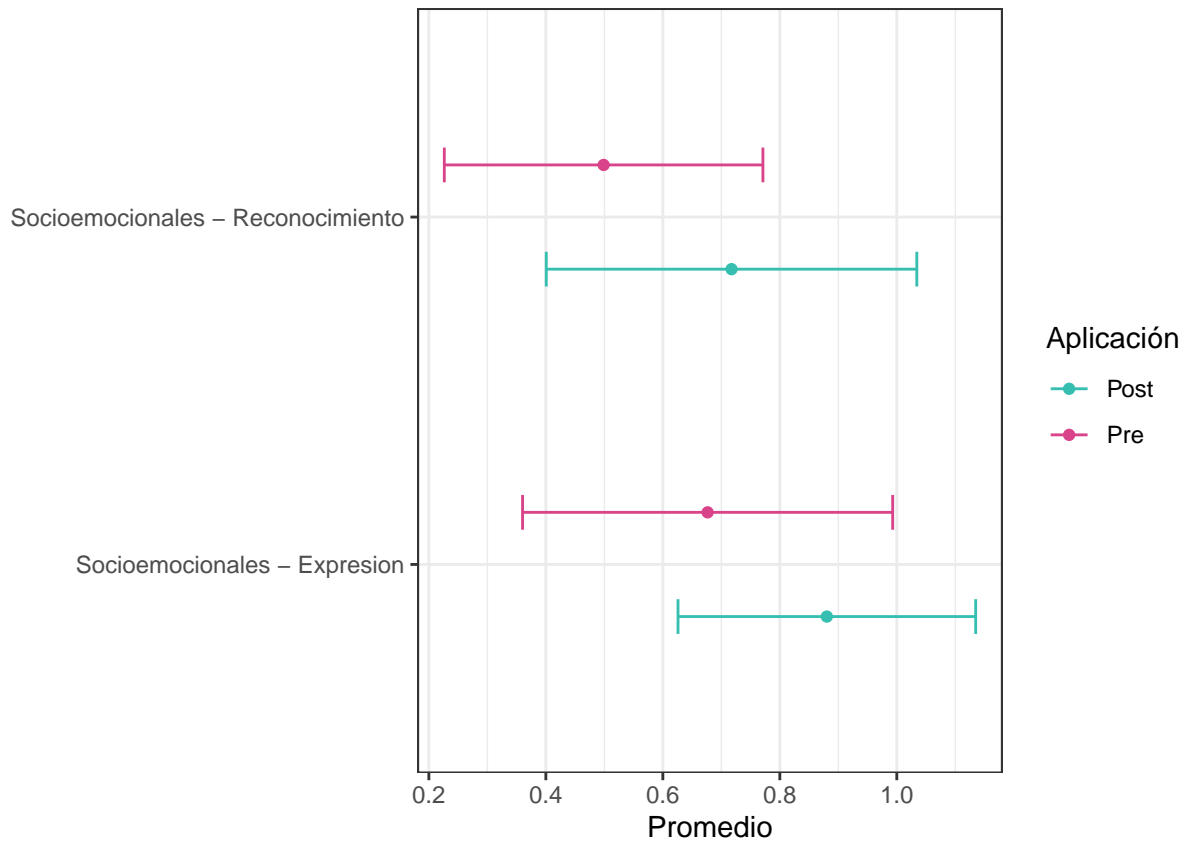
```
# Guardamos la imagen
```

```
ggsave("../Plots/primer_aaplicacion/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/primer_aPLICACION/significativas_socioemocionales.png")
```

4.4.6 Diferencias entre ciudades

```
base = c("Código" , "Ciudad")
```

```
data_ciudades = dplyr::select(actitudes_post, base)
```

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

```
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))
```

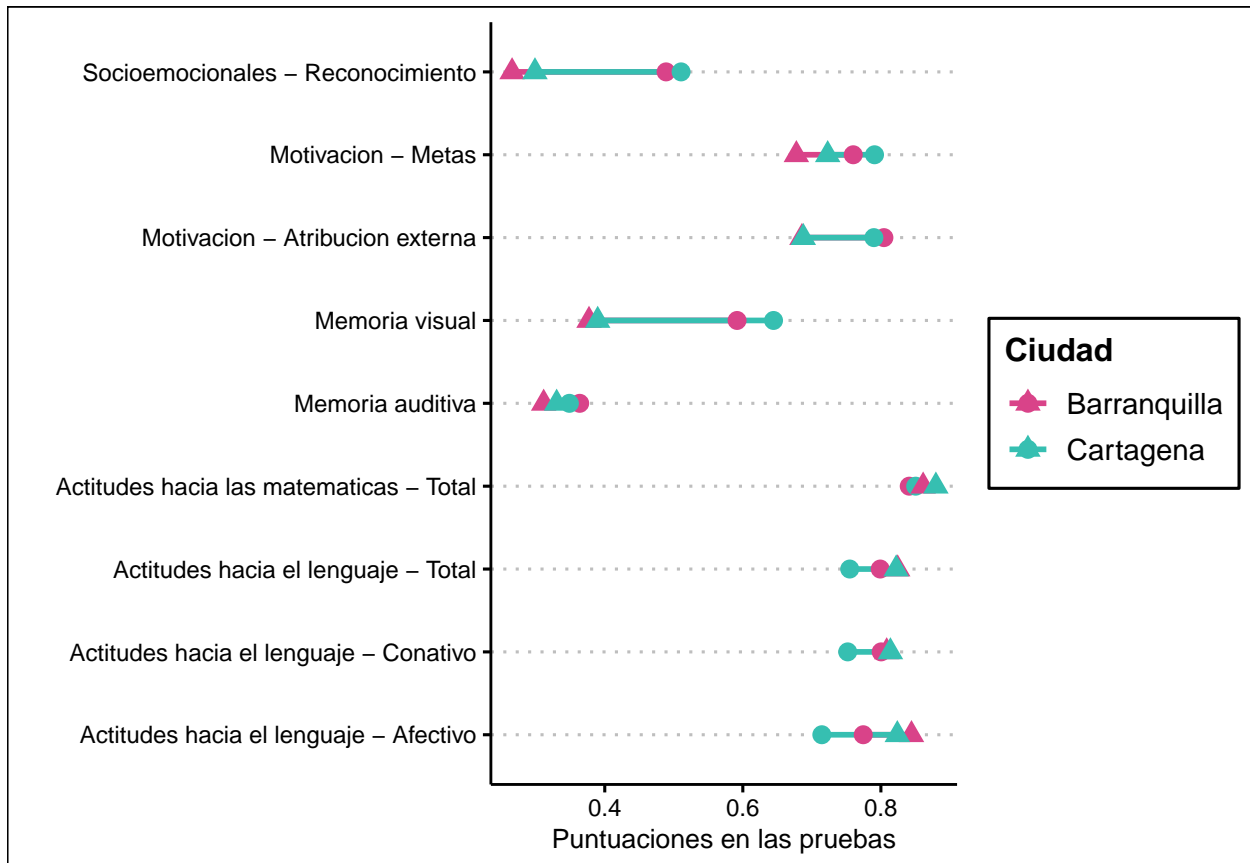
```
pivot_ciudades = dplyr::left_join(pivot_final, data_ciudades, by = "Código")
```

```

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/primer_aaplicacion/significativas_ciudades.png")
```

4.4.7 Diferencias entre ciclos

```
base = c("Código" , "Ciclo")

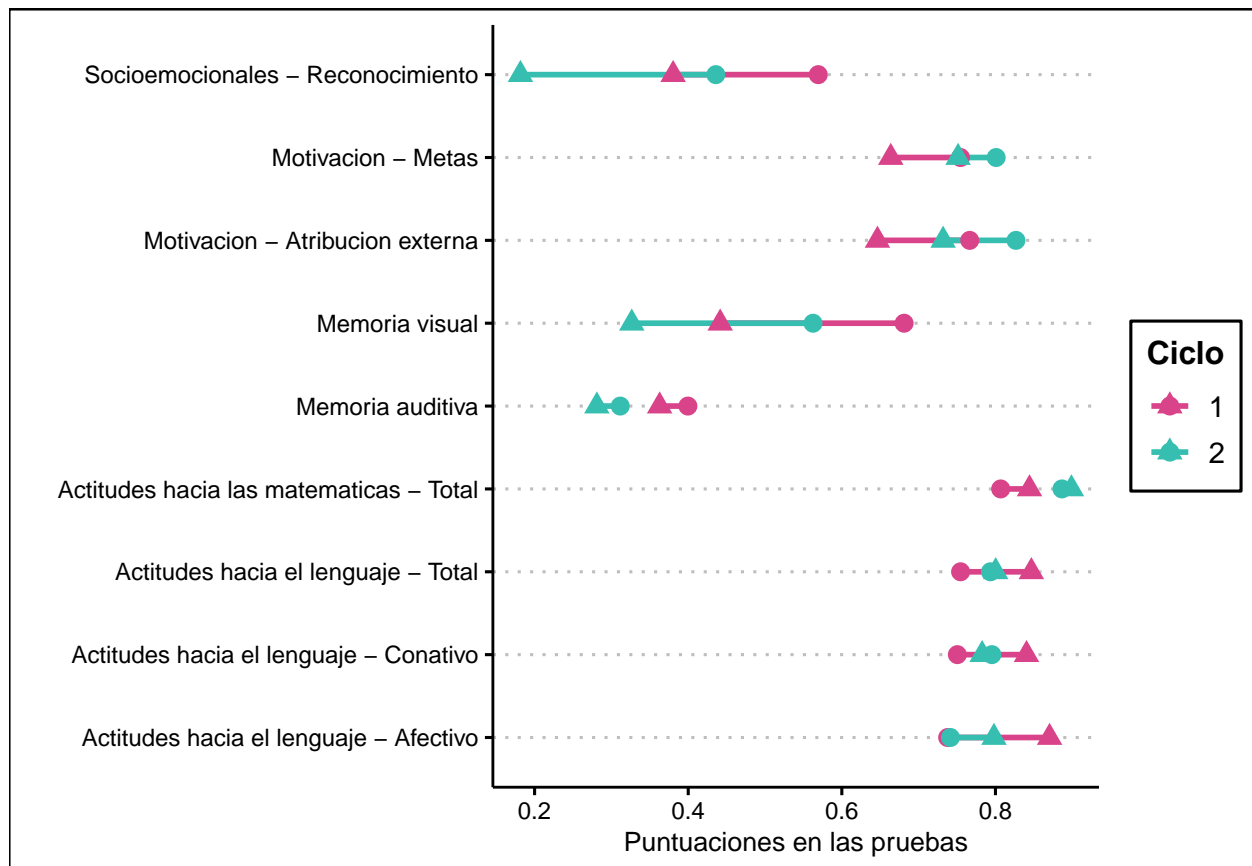
data_ciclos = dplyr::select(actitudes_post, base)
data_ciclos = rbind(data_ciclos, dplyr::select(socioemocional_post, base))
data_ciclos = rbind(data_ciclos, dplyr::select(motivacion_post, base))
data_ciclos = rbind(data_ciclos, dplyr::select(inhibicion_post, base))
data_ciclos = rbind(data_ciclos, dplyr::select(mem_vis_post, 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/primer_aaplicacion/significativas_ciclos.png")
```

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)
