

# Resultados poblacionales

## Generación de resultados

Carrasco, D., PhD & Miranda, D., PhD

Centro de Medición MIDE UC

Castillo, C., Mg.

Estudiante de Doctorado Educación UC

LLECE: Taller de Análisis III

Santiago, Marzo 02 de 2022

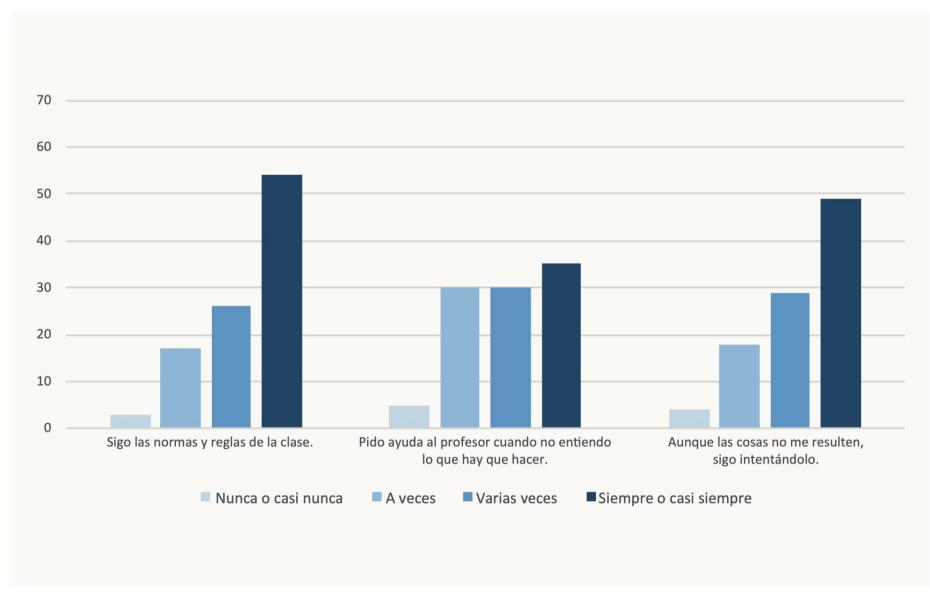
Poblacionales

# Generación de resultados poblaciones

Reproducido resultados de ERCE 2019

## Resultados poblaciones

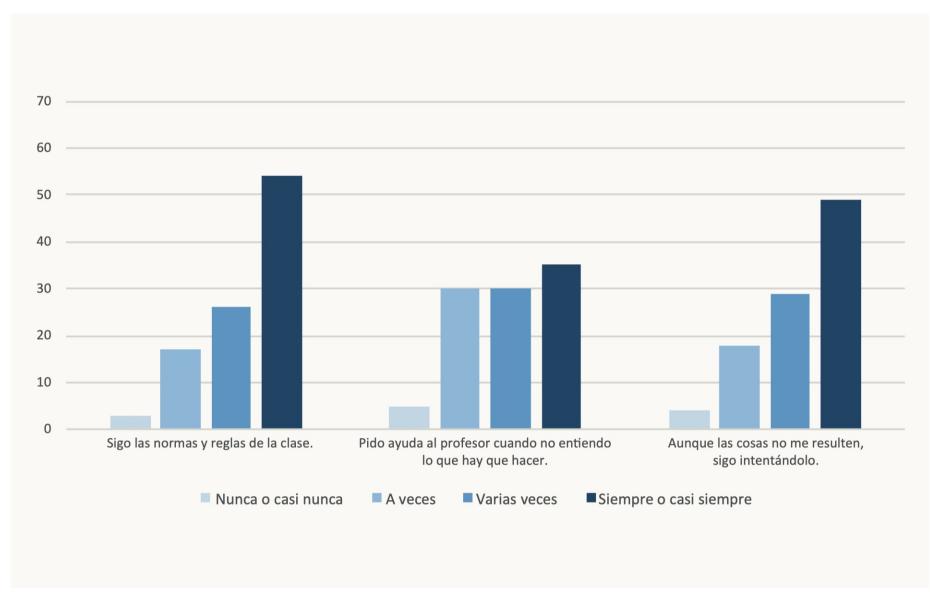
Figura 5. Distribución de respuestas a nivel regional para tres ítems de la escala de autorregulación escolar.



**Nota:** Figura 5 del informe "Habilidades socioemocionales en América Latina y el Caribe" (UNESCO-OREALC, 2021a). Descriptivo de ítems de la escala de 'Autorregulación Escolar'. Ver código [05\\_resultados\\_codigos.rmd](#).

## Resultados poblaciones

Figura 5. Distribución de respuestas a nivel regional para tres ítems de la escala de autorregulación escolar.

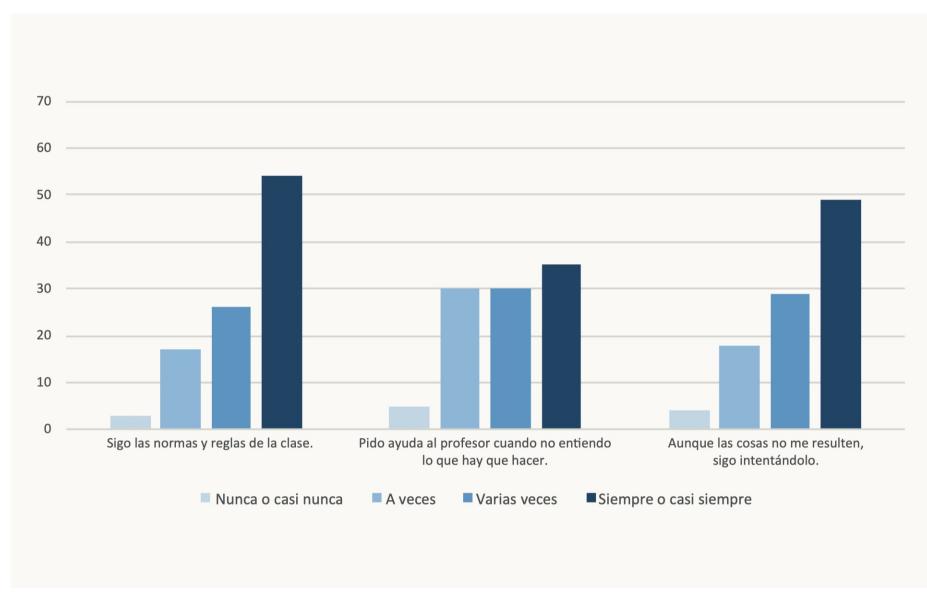


- Llamaremos **resultados poblacionales** a aquellas cifras que refieren a la población de estudiantes, ya sea de un país, o de toda la región.

**Nota:** Figura 5 del informe "Habilidades socioemocionales en América Latina y el Caribe" (UNESCO-OREALC, 2021a). Descriptivo de ítems de la escala de 'Autorregulación Escolar'. Ver código [05\\_resultados\\_codigos.rmd](#).

## Resultados poblaciones

Figura 5. Distribución de respuestas a nivel regional para tres ítems de la escala de autorregulación escolar.

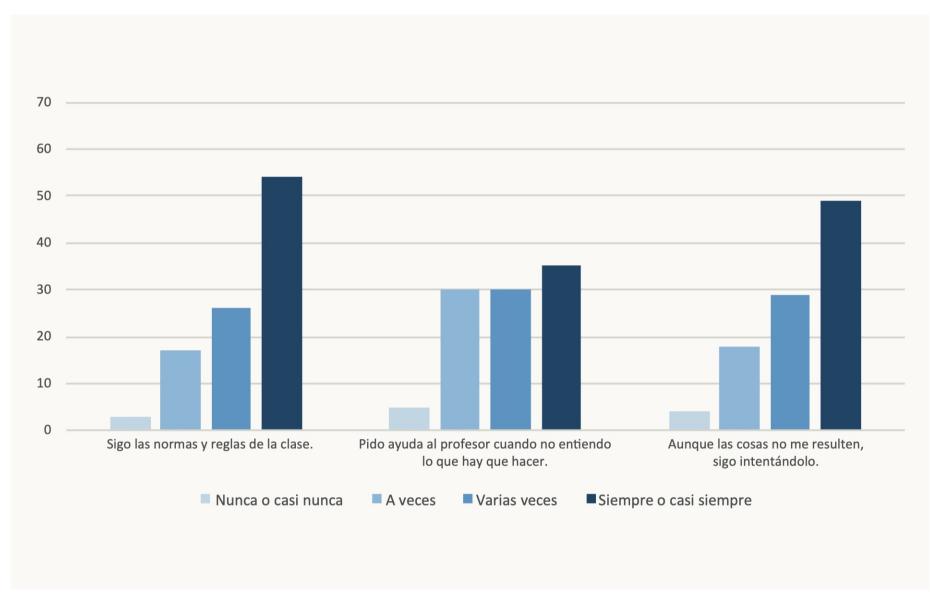


- Llamaremos **resultados poblacionales** a aquellas cifras que refieren a la población de estudiantes, ya sea de un país, o de toda la región.
- En el siguiente ejemplo, ocupamos la **Figura 5** del informe de Habilidades Socioemocionales (UNESCO-OREALC, 2021a, p26). Esta figura grafica los porcentajes de respuesta esperados para toda la región, en tres ítems de la escala **Autorregulación escolar**

**Nota:** Figura 5 del informe "Habilidades socioemocionales en América Latina y el Caribe" (UNESCO-OREALC, 2021a). Descriptivo de ítems de la escala de 'Autorregulación Escolar'. Ver código [05\\_resultados\\_codigos.rmd](#).

## Resultados poblaciones

Figura 5. Distribución de respuestas a nivel regional para tres ítems de la escala de autorregulación escolar.



**Nota:** Figura 5 del informe "Habilidades socioemocionales en América Latina y el Caribe" (UNESCO-OREALC, 2021a). Descriptivo de ítems de la escala de 'Autorregulación Escolar'. Ver código [05\\_resultados\\_codigos.rmd](#).

- Llamaremos **resultados poblacionales** a aquellas cifras que refieren a la población de estudiantes, ya sea de un país, o de toda la región.
- En el siguiente ejemplo, ocupamos la **Figura 5** del informe de Habilidades Socioemocionales (UNESCO-OREALC, 2021a, p26). Esta figura grafica los porcentajes de respuesta esperados para toda la región, en tres ítems de la escala **Autorregulación escolar**
- Para reproducir estos resultados, siguiendo la línea de **análisis como desarrollo**, necesitamos generar un código que incluya una serie de pasos, desde la apertura de los datos, la identificación de las variables de interés, hasta la presentación de resultados.
- A continuación, revisaremos los principales pasos para reproducir los resultados de la Figura 5., como un todo. Y luego, revisaremos estos pasos uno a uno.

Poblacionales

# Cálculo de porcentajes

Reproducido resultados de porcentajes (UNESCO-OREALC, 2021a, Figura 5)

```

# figura 5
#
# abrir datos
#-
erce_qa6 <- erce::erce_2019_qa6 %>%
  erce::remove_labels()
erce_hse <- erce::erce_2019_hse %>%
  erce::remove_labels()

# inspeccionar datos-hse
#-
# seleccionar datos
#-
erce_hse_selected <- erce_hse %>%
  dplyr::select(-cluster)
  # clustering
  # IDCNTRY, ID SCHOOL, ID CLASS, ID STUD,
  # items
  MS3MR1:VD3A06
  )

# unir datos y crear cluster únicos
#-
erce_a6 <- erce_qa6 %>%
  dplyr::left_join(erce_hse_selected,
    by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')) %>%
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, " ", STRATA)))) %>%
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, " ", IDSCHOOL)))) %>%
  mutate(id_i = seq(1:nrow(.)))

# base de datos con diseño
#-
library(survey)
erce_tsl <- erce_a6 %>%
  as_survey_design(
    strata = id_s,
    ids = id_i,
    weights = ws,
    nest = TRUE)

# Opción: corrección a unidad primaria de muestreo que resulte
# única al estrato
library(survey)
options(survey.lonely.psu="adjust")

#-
# ítems de interés
#-
# MS3MR1
# Sigo las normas y reglas de la clase.
# MS3MR2
# Pido ayuda al profesor cuando no entiendo lo que hay que hacer.
# MS3PP2
# Aunque una tarea sea muy difícil, sigo trabajando en ella.
#-
# tabla de porcentajes para un ítem
#-
erce_tsl %>%
  summarize(
    p1 = survey_mean(MS3MR1==1, na.rm = TRUE),
    p2 = survey_mean(MS3MR1==2, na.rm = TRUE),
    p3 = survey_mean(MS3MR1==3, na.rm = TRUE),
    p4 = survey_mean(MS3MR1==4, na.rm = TRUE),
    s1 = unweighted(mean(MS3MR1==1, na.rm = TRUE)),
    s2 = unweighted(mean(MS3MR1==2, na.rm = TRUE)),
    s3 = unweighted(mean(MS3MR1==3, na.rm = TRUE)),
    s4 = unweighted(mean(MS3MR1==4, na.rm = TRUE)))
  ) %>%
  mutate(item = 'MS3MR1') %>%
  dplyr::select(item, p1, p2, p3, p4) %>%
  knitr::kable(digits = 2)

# Nota: aunque calculamos errores estandar, y
# puntos estimados sin diseño, por motivos
# ilustrativos, solo seleccionamos los puntos
# estimados generalizables a la región.

```

## Poblacionales: Cálculo de porcentajes (porcentajes de respuesta para la región)

- A la izquierda tenemos la secuencia completa para poder obtener resultados parciales de la figura 5. En particular, los resultados al ítem **MS3MR1**.

```

# figura 5
#
# abrir datos
#-
erce_qa6 <- erce::erce_2019_qa6 %>%
  erce::remove_labels()
erce_hse <- erce::erce_2019_hse %>%
  erce::remove_labels()
# inspeccionar datos-hse
#-
dplyr::glimpse(erce_hse)
#-
# seleccionar datos
#-
erce_hse_selected <- erce_hse %>%
  dplyr::select(-
    # clustering
    IDCNTRY, IDSCHOOL, IDCLASS, IDSTUD,
    # items
    MS3MR1:VD3A06
  )
#-
# unir datos y crear cluster únicos
#-
erce_a6 <- erce_qa6 %>%
  dplyr::left_join(., erce_hse_selected,
    by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')) %>%
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, "_", STRATA)))) %>%
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, "_", IDSCHOOL)))) %>%
  mutate(id_i = seq(1:nrow(.)))
#-
# base de datos con diseño
#-
# survey method: taylor series linearization library(srvyr)
library(srvyr)
erce_tsl <- erce_a6 %>%
  as_survey_design(
    strata = id_s,
    ids = id_i,
    weights = ws,
    nest = TRUE)
# Opción: corrección a unidad primaria de muestreo que resulte
# única al estrato
library(survey)
options(survey.lonely.psu="adjust")
#-
# ítems de interés
#-
# MS3MR1
# Sigo las normas y reglas de la clase.
# MS3MR2
# Pido ayuda al profesor cuando no entiendo lo que hay que hacer.
# MS3PP2
# Aunque una tarea sea muy difícil, sigo trabajando en ella.
#-
# tabla de porcentajes para un ítem
#-
erce_tsl %>%
summarize(
  p1 = survey_mean(MS3MR1==1, na.rm = TRUE),
  p2 = survey_mean(MS3MR1==2, na.rm = TRUE),
  p3 = survey_mean(MS3MR1==3, na.rm = TRUE),
  p4 = survey_mean(MS3MR1==4, na.rm = TRUE),
  s1 = unweighted(mean(MS3MR1==1, na.rm = TRUE)),
  s2 = unweighted(mean(MS3MR1==2, na.rm = TRUE)),
  s3 = unweighted(mean(MS3MR1==3, na.rm = TRUE)),
  s4 = unweighted(mean(MS3MR1==4, na.rm = TRUE)))
) %>%
mutate(item = 'MS3MR1') %>%
dplyr::select(item, p1, p2, p3, p4) %>%
knitr::kable(., digits = 2)

# Nota: aunque calculamos errores estandar, y
# puntos estimados sin diseño, por motivos
# ilustrativos, solo seleccionamos los puntos
# estimados generalizables a la región.

```

## Poblacionales: Cálculo de porcentajes (porcentajes de respuesta para la región)

- A la izquierda tenemos la secuencia completa para poder obtener resultados parciales de la figura 5. En particular, los resultados al ítem **MS3MR1**.
- Primero necesitamos **cargar los datos** de Cuestionario de Sexto Grado (**erce::erce\_2019\_qa6**), y los datos de cuestionario del módulo de habilidades socioemocionales (**erce::erce\_2019\_hse**).

```

# figura 5
#
# abrir datos
#---
erce_qa6 <- erce::erce_2019_qa6 %>%
  erce::remove_labels()
erce_hse <- erce::erce_2019_hse %>%
  erce::remove_labels()
# inspeccionar datos-hse
#---
dplyr::glimpse(erce_hse)
#---
# seleccionar datos
#---
erce_hse_selected <- erce_hse %>%
  dplyr::select(
    # clustering
    IDCNTRY, IDSCHOOL, IDCLASS, IDSTUD,
    # items
    MS3MR1:VD3A06
  )
# unir datos y crear cluster únicos
#---
erce_a6 <- erce_qa6 %>%
  dplyr::left_join(., erce_hse_selected,
  by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')) %>%
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, "_", STRATA)))) %>%
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, "_", IDSCHOOL)))) %>%
  mutate(id_i = seq(1:nrow(.)))
# base de datos con diseño
#---
# survey method: taylor series linearization library(srvyr)
library(srvyr)
erce_tsl <- erce_a6 %>%
  as_survey_design(
    strata = id_s,
    ids = id_i,
    weights = ws,
    nest = TRUE)
# Opción: corrección a unidad primaria de muestreo que resulte
# única al estrato
library(survey)
options(survey.lonely.psu="adjust")
#---
# ítems de interés
#---
# MS3MR1
# Sigo las normas y reglas de la clase.
# MS3MR2
# Pido ayuda al profesor cuando no entiendo lo que hay que hacer.
# MS3PP2
# Aunque una tarea sea muy difícil, sigo trabajando en ella.
#---
# tabla de porcentajes para un ítem
#---
erce_tsl %>%
  summarize(
    p1 = survey_mean(MS3MR1==1, na.rm = TRUE),
    p2 = survey_mean(MS3MR1==2, na.rm = TRUE),
    p3 = survey_mean(MS3MR1==3, na.rm = TRUE),
    p4 = survey_mean(MS3MR1==4, na.rm = TRUE),
    s1 = unweighted(mean(MS3MR1==1, na.rm = TRUE)),
    s2 = unweighted(mean(MS3MR1==2, na.rm = TRUE)),
    s3 = unweighted(mean(MS3MR1==3, na.rm = TRUE)),
    s4 = unweighted(mean(MS3MR1==4, na.rm = TRUE))
  ) %>%
  mutate(item = 'MS3MR1') %>%
  dplyr::select(item, p1, p2, p3, p4) %>%
  knitr::kable(digits = 2)

# Nota: aunque calculamos errores estandar, y
# puntos estimados sin diseño, por motivos
# ilustrativos, solo seleccionamos los puntos
# estimados generalizables a la región.

```

## Poblacionales: Cálculo de porcentajes (porcentajes de respuesta para la región)

- A la izquierda tenemos la secuencia completa para poder obtener resultados parciales de la figura 5. En particular, los resultados al ítem **MS3MR1**.
- Primero necesitamos **cargar los datos** de Cuestionario de Sexto Grado (**erce::erce\_2019\_qa6**), y los datos de cuestionario del módulo de habilidades socioemocionales (**erce::erce\_2019\_hse**).
- Luego de cargados los datos, necesitamos **ver el contenido** de los datos. En particular, queremos inspeccionar los datos del módulo de habilidades socioemocionales, porque queremos recuperar información específica de esta base de datos. Empleamos la función **dplyr::glimpse** para este cometido, luego de haber removido la meta data con **erce::remove\_labels**.

```

# figura 5
#
# abrir datos
#---
erce_qa6 <- erce::erce_2019_qa6 %>%
  erce::remove_labels()
erce_hse <- erce::erce_2019_hse %>%
  erce::remove_labels()
# inspeccionar datos-hse
#---
dplyr::glimpse(erce_hse)
#---
# seleccionar datos
#---
erce_hse_selected <- erce_hse %>%
  dplyr::select(
    # clustering
    IDCNTRY, ID SCHOOL, ID CLASS, ID STUD,
    # items
    MS3MR1:VD3A06
  )
# unir datos y crear cluster únicos
#---
erce_a6 <- erce_qa6 %>%
  dplyr::left_join(., erce_hse_selected,
  by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')) %>%
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, "_", STRATA)))) %>%
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, "_", IDSCHOOL)))) %>%
  mutate(id_i = seq(1:nrow(.)))
# base de datos con diseño
#---

# survey method: taylor series linearization library(srvyr)
library(srvyr)
erce_tsl <- erce_a6 %>%
  as_survey_design(
    strata = id_s,
    ids = id_i,
    weights = ws,
    nest = TRUE)
# Opción: corrección a unidad primaria de muestreo que resulte
# única al estrato
library(survey)
options(survey.lonely.psu="adjust")
#---
# ítems de interés
#---
# MS3MR1
# Sigo las normas y reglas de la clase.
# MS3MR2
# Pido ayuda al profesor cuando no entiendo lo que hay que hacer.
# MS3PP2
# Aunque una tarea sea muy difícil, sigo trabajando en ella.
#---
# tabla de porcentajes para un ítem
#---
erce_tsl %>%
  summarize(
    p1 = survey_mean(MS3MR1==1, na.rm = TRUE),
    p2 = survey_mean(MS3MR1==2, na.rm = TRUE),
    p3 = survey_mean(MS3MR1==3, na.rm = TRUE),
    p4 = survey_mean(MS3MR1==4, na.rm = TRUE),
    s1 = unweighted(mean(MS3MR1==1, na.rm = TRUE)),
    s2 = unweighted(mean(MS3MR1==2, na.rm = TRUE)),
    s3 = unweighted(mean(MS3MR1==3, na.rm = TRUE)),
    s4 = unweighted(mean(MS3MR1==4, na.rm = TRUE))
  ) %>%
  mutate(item = 'MS3MR1') %>%
  dplyr::select(item, p1, p2, p3, p4) %>%
  knitr::kable(digits = 2)

# Nota: aunque calculamos errores estandar, y
# puntos estimados sin diseño, por motivos
# ilustrativos, solo seleccionamos los puntos
# estimados generalizables a la región.

```

## Poblacionales: Cálculo de porcentajes (porcentajes de respuesta para la región)

- A la izquierda tenemos la secuencia completa para poder obtener resultados parciales de la figura 5. En particular, los resultados al ítem **MS3MR1**.
- Primero necesitamos **cargar los datos** de Cuestionario de Sexto Grado (**erce::erce\_2019\_qa6**), y los datos de cuestionario del módulo de habilidades socioemocionales (**erce::erce\_2019\_hse**).
- Luego de cargados los datos, necesitamos **ver el contenido** de los datos. En particular, queremos inspeccionar los datos del módulo de habilidades socioemocionales, porque queremos recuperar información específica de esta base de datos. Empleamos la función **dplyr::glimpse** para este cometido, luego de haber removido la meta data con **erce::remove\_labels**.
- Lo que queremos es **seleccionar** los datos específicos que queremos, los cuales son los datos de clustering, y las respuestas a los ítems. Aquí empleamos la función **dplyr::select()** para cumplir este propósito.

```

# figura 5
#
# abrir datos
#---
erce_qa6 <- erce::erce_2019_qa6 %>%
  erce::remove_labels()
erce_hse <- erce::erce_2019_hse %>%
  erce::remove_labels()
# inspeccionar datos-hse
#---
dplyr::glimpse(erce_hse)
#---
# seleccionar datos
#---
erce_hse_selected <- erce_hse %>%
  dplyr::select(
    # clustering
    IDCTRY, IDSCHOOL, IDCLASS, IDSTUD,
    # items
    MS3MR1:VD3A06
  )
# unir datos y crear cluster únicos
#---
erce_a6 <- erce_qa6 %>%
  dplyr::left_join(.., erce_hse_selected,
  by = c('IDCTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')) %>%
  mutate(id_k = as.numeric(as.factor(paste0(IDCTRY)))) %>%
  mutate(id_s = as.numeric(as.factor(paste0(IDCTRY, "_", STRATA)))) %>%
  mutate(id_j = as.numeric(as.factor(paste0(IDCTRY, "_", IDSCHOOL)))) %>%
  mutate(id_i = seq(1:nrow(.)))
# base de datos con diseño
#---
# survey method: taylor series linearization library(srvyr)
library(srvyr)
erce_tsl <- erce_a6 %>%
  as_survey_design(
    strata = id_s,
    ids = id_i,
    weights = ws,
    nest = TRUE)
# Opción: corrección a unidad primaria de muestreo que resulte
# única al estrato
library(survey)
options(survey.lonely.psu="adjust")
#---
# items de interés
#---
# MS3MR1
# Sigo las normas y reglas de la clase.
# MS3MR2
# Pido ayuda al profesor cuando no entiendo lo que hay que hacer.
# MS3PP2
# Aunque una tarea sea muy difícil, sigo trabajando en ella.
#---
# tabla de porcentajes para un ítem
#---
erce_tsl %>%
  summarize(
    p1 = survey_mean(MS3MR1==1, na.rm = TRUE),
    p2 = survey_mean(MS3MR1==2, na.rm = TRUE),
    p3 = survey_mean(MS3MR1==3, na.rm = TRUE),
    p4 = survey_mean(MS3MR1==4, na.rm = TRUE),
    s1 = unweighted(mean(MS3MR1==1, na.rm = TRUE)),
    s2 = unweighted(mean(MS3MR1==2, na.rm = TRUE)),
    s3 = unweighted(mean(MS3MR1==3, na.rm = TRUE)),
    s4 = unweighted(mean(MS3MR1==4, na.rm = TRUE))
  ) %>%
  mutate(item = 'MS3MR1') %>%
  dplyr::select(item, p1, p2, p3, p4) %>%
  knitr::kable(digits = 2)

# Nota: aunque calculamos errores estandar, y
# puntos estimados sin diseño, por motivos
# ilustrativos, solo seleccionamos los puntos
# estimados generalizables a la región.

```

## Poblacionales: Cálculo de porcentajes (porcentajes de respuesta para la región)

- A la izquierda tenemos la secuencia completa para poder obtener resultados parciales de la figura 5. En particular, los resultados al ítem **MS3MR1**.
- Primero necesitamos **cargar los datos** de Cuestionario de Sexto Grado (**erce::erce\_2019\_qa6**), y los datos de cuestionario del módulo de habilidades socioemocionales (**erce::erce\_2019\_hse**).
- Luego de cargados los datos, necesitamos **ver el contenido** de los datos. En particular, queremos inspeccionar los datos del módulo de habilidades socioemocionales, porque queremos recuperar información específica de esta base de datos. Empleamos la función **dplyr::glimpse** para este cometido, luego de haber removido la meta data con **erce::remove\_labels**.
- Lo que queremos es **seleccionar** los datos específicos que queremos, los cuales son los datos de clustering, y las respuestas a los ítems. Aquí empleamos la función **dplyr::select()** para cumplir este propósito.
- Como hemos seleccionado de forma específica a las variables que queremos unir al cuestionario de estudiantes, nos estamos asegurando de no generar variables repetidas luego de la unión de bases de datos. Para unir los datos empleamos la función **dplyr::left\_join**, la cual nos permite agregar a la base de datos de Estudiantes de Sexto grado, las respuestas de cuestionario de Habilidades Socioemocionales.

```

# figura 5
#
# abrir datos
#---
erce_qa6 <- erce::erce_2019_qa6 %>%
  erce::remove_labels()
erce_hse <- erce::erce_2019_hse %>%
  erce::remove_labels()
# inspeccionar datos-hse
#---
dplyr::glimpse(erce_hse)
#-
# seleccionar datos
#---
erce_hse_selected <- erce_hse %>%
  dplyr::select(
    # clustering
    IDCNTRY, ID SCHOOL, IDCLASS, IDSTUD,
    # items
    MS3MR1:VD3A06
  )
#-
# unir datos y crear cluster únicos
#---
erce_a6 <- erce_qa6 %>%
  dplyr::left_join(., erce_hse_selected,
  by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')) %>%
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, "_", STRATA)))) %>%
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, "_", IDSCHOOL)))) %>%
  mutate(id_i = seq(1:nrow(.)))
#-
# base de datos con diseño
#---
# survey method: taylor series linearization library(srvyr)
library(srvyr)
erce_tsl <- erce_a6 %>%
  as_survey_design(
    strata = id_s,
    ids = id_i,
    weights = ws,
    nest = TRUE)
# Opción: corrección a unidad primaria de muestreo que resulte
# única al estrato
library(survey)
options(survey.lonely.psu="adjust")
#-
# items de interés
#---
# MS3MR1
# Sigo las normas y reglas de la clase.
# MS3MR2
# Pido ayuda al profesor cuando no entiendo lo que hay que hacer.
# MS3PP2
# Aunque una tarea sea muy difícil, sigo trabajando en ella.
#-
# tabla de porcentajes para un ítem
#---
erce_tsl %>%
  summarize(
    p1 = survey_mean(MS3MR1==1, na.rm = TRUE),
    p2 = survey_mean(MS3MR1==2, na.rm = TRUE),
    p3 = survey_mean(MS3MR1==3, na.rm = TRUE),
    p4 = survey_mean(MS3MR1==4, na.rm = TRUE),
    s1 = unweighted(mean(MS3MR1==1, na.rm = TRUE)),
    s2 = unweighted(mean(MS3MR1==2, na.rm = TRUE)),
    s3 = unweighted(mean(MS3MR1==3, na.rm = TRUE)),
    s4 = unweighted(mean(MS3MR1==4, na.rm = TRUE))
  ) %>%
  mutate(item = 'MS3MR1') %>%
  dplyr::select(item, p1, p2, p3, p4) %>%
  knitr::kable(., digits = 2)

# Nota: aunque calculamos errores estandar, y
# puntos estimados sin diseño, por motivos
# ilustrativos, solo seleccionamos los puntos
# estimados generalizables a la región.

```

## Poblacionales: Cálculo de porcentajes (porcentajes de respuesta para la región)

- A la izquierda tenemos la secuencia completa para poder obtener resultados parciales de la figura 5. En particular, los resultados al ítem **MS3MR1**.
- Primero necesitamos **cargar los datos** de Cuestionario de Sexto Grado (**erce::erce\_2019\_qa6**), y los datos de cuestionario del módulo de habilidades socioemocionales (**erce::erce\_2019\_hse**).
- Luego de cargados los datos, necesitamos **ver el contenido** de los datos. En particular, queremos inspeccionar los datos del módulo de habilidades socioemocionales, porque queremos recuperar información específica de esta base de datos. Empleamos la función **dplyr::glimpse** para este cometido, luego de haber removido la meta data con **erce::remove\_labels**.
- Lo que queremos es **seleccionar** los datos específicos que queremos, los cuales son los datos de clustering, y las respuestas a los ítems. Aquí empleamos la función **dplyr::select()** para cumplir este propósito.
- Como hemos seleccionado de forma específica a las variables que queremos unir al cuestionario de estudiantes, nos estamos asegurando de no generar variables repetidas luego de la unión de bases de datos. Para unir los datos empleamos la función **dplyr::left\_join**, la cual nos permite agregar a la base de datos de Estudiantes de Sexto grado, las respuestas de cuestionario de Habilidades Socioemocionales.
- En la unión de base de datos, es importante que empleemos las variables de clustering completas. En este caso, estamos empleando la siguiente línea de código: **by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')**.

```

# figura 5
#
# abrir datos
#---
erce_qa6 <- erce::erce_2019_qa6 %>%
  erce::remove_labels()
erce_hse <- erce::erce_2019_hse %>%
  erce::remove_labels()
# inspeccionar datos-hse
#---
dplyr::glimpse(erce_hse)
#---
# seleccionar datos
#---
erce_hse_selected <- erce_hse %>%
  dplyr::select(-
    # clustering
    IDCNTRY, ID SCHOOL, IDCLASS, IDSTUD,
    # items
    MS3MR1:VD3A06
  )
# unir datos y crear cluster únicos
#---
erce_a6 <- erce_qa6 %>%
  dplyr::left_join(., erce_hse_selected,
    by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')) %>%
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, "_", STRATA)))) %>%
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, "_", IDSCHOOL)))) %>%
  mutate(id_i = seq(1:nrow(.)))
# base de datos con diseño
#---
# survey method: taylor series linearization library(srvyr)
library(srvyr)
erce_tsl <- erce_a6 %>%
  as_survey_design(
    strata = id_s,
    ids = id_i,
    weights = ws,
    nest = TRUE)
# Opción: corrección a unidad primaria de muestreo que resulte
# única al estrato
library(survey)
options(survey.lonely.psu="adjust")
#---
# ítems de interés
#---
# MS3MR1
# Sigo las normas y reglas de la clase.
# MS3MR2
# Pido ayuda al profesor cuando no entiendo lo que hay que hacer.
# MS3PP2
# Aunque una tarea sea muy difícil, sigo trabajando en ella.
#---
# tabla de porcentajes para un ítem
#---
erce_tsl %>%
summarize(
  p1 = survey_mean(MS3MR1==1, na.rm = TRUE),
  p2 = survey_mean(MS3MR1==2, na.rm = TRUE),
  p3 = survey_mean(MS3MR1==3, na.rm = TRUE),
  p4 = survey_mean(MS3MR1==4, na.rm = TRUE),
  s1 = unweighted(mean(MS3MR1==1, na.rm = TRUE)),
  s2 = unweighted(mean(MS3MR1==2, na.rm = TRUE)),
  s3 = unweighted(mean(MS3MR1==3, na.rm = TRUE)),
  s4 = unweighted(mean(MS3MR1==4, na.rm = TRUE)))
) %>%
mutate(item = 'MS3MR1') %>%
dplyr::select(item, p1, p2, p3, p4) %>%
knitr::kable(., digits = 2)

# Nota: aunque calculamos errores estandar, y
# puntos estimados sin diseño, por motivos
# ilustrativos, solo seleccionamos los puntos
# estimados generalizables a la región.

```

## Poblacionales: Cálculo de porcentajes (porcentajes de respuesta para la región)

- A la izquierda tenemos la secuencia completa para poder obtener resultados parciales de la figura 5. En particular, los resultados al ítem **MS3MR1**.
- Primero necesitamos **cargar los datos** de Cuestionario de Sexto Grado (**erce::erce\_2019\_qa6**), y los datos de cuestionario del módulo de habilidades socioemocionales (**erce::erce\_2019\_hse**).
- Luego de cargados los datos, necesitamos **ver el contenido** de los datos. En particular, queremos inspeccionar los datos del módulo de habilidades socioemocionales, porque queremos recuperar información específica de esta base de datos. Empleamos la función **dplyr::glimpse** para este cometido, luego de haber removido la meta data con **erce::remove\_labels**.
- Lo que queremos es **seleccionar** los datos específicos que queremos, los cuales son los datos de clustering, y las respuestas a los ítems. Aquí empleamos la función **dplyr::select()** para cumplir este propósito.
- Como hemos seleccionado de forma específica a las variables que queremos unir al cuestionario de estudiantes, nos estamos asegurando de no generar variables repetidas luego de la unión de bases de datos. Para unir los datos empleamos la función **dplyr::left\_join**, la cual nos permite agregar a la base de datos de Estudiantes de Sexto grado, las respuestas de cuestionario de Habilidades Socioemocionales.
- En la unión de base de datos, es importante que empleemos las variables de clustering completas. En este caso, estamos empleando la siguiente línea de código: **by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')**.
- Con los datos unidos, creamos una base de datos con **Taylor Series Linearization** (ver Heeringa et al., 2009), empleando a la librería **srvyr**. Empleamos este método de cálculo de errores, porque es más rápido que BRR para producir resultados, y produce resultados muy similares a los que genera BRR.

```

# figura 5
#
# abrir datos
#---
erce_qa6 <- erce::erce_2019_qa6 %>%
  erce::remove_labels()
erce_hse <- erce::erce_2019_hse %>%
  erce::remove_labels()
# inspeccionar datos-hse
#---
dplyr::glimpse(erce_hse)
#---
# seleccionar datos
#---
erce_hse_selected <- erce_hse %>%
  dplyr::select(
    # clustering
    IDCNTRY, ID SCHOOL, IDCLASS, IDSTUD,
    # items
    MS3MR1:VD3A06
  )
# unir datos y crear cluster únicos
#---
erce_a6 <- erce_qa6 %>%
  dplyr::left_join(., erce_hse_selected,
  by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')) %>%
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, "_", STRATA)))) %>%
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, "_", IDSCHOOL)))) %>%
  mutate(id_i = seq(1:nrow(.)))
# base de datos con diseño
#---
# survey method: taylor series linearization library(srvyr)
library(srvyr)
erce_tsl <- erce_a6 %>%
  as_survey_design(
    strata = id_s,
    ids = id_i,
    weights = ws,
    nest = TRUE)
# Opción: corrección a unidad primaria de muestreo que resulte
# única al estrato
library(survey)
options(survey.lonely.psu="adjust")
#---
# ítems de interés
#---
# MS3MR1
# Sigo las normas y reglas de la clase.
# MS3MR2
# Pido ayuda al profesor cuando no entiendo lo que hay que hacer.
# MS3PP2
# Aunque una tarea sea muy difícil, sigo trabajando en ella.
#---
# tabla de porcentajes para un ítem
#---
erce_tsl %>%
summarize(
  p1 = survey_mean(MS3MR1==1, na.rm = TRUE),
  p2 = survey_mean(MS3MR1==2, na.rm = TRUE),
  p3 = survey_mean(MS3MR1==3, na.rm = TRUE),
  p4 = survey_mean(MS3MR1==4, na.rm = TRUE),
  s1 = unweighted(mean(MS3MR1==1, na.rm = TRUE)),
  s2 = unweighted(mean(MS3MR1==2, na.rm = TRUE)),
  s3 = unweighted(mean(MS3MR1==3, na.rm = TRUE)),
  s4 = unweighted(mean(MS3MR1==4, na.rm = TRUE)))
) %>%
mutate(item = 'MS3MR1') %>%
dplyr::select(item, p1, p2, p3, p4) %>%
knitr::kable(., digits = 2)

# Nota: aunque calculamos errores estandar, y
# puntos estimados sin diseño, por motivos
# ilustrativos, solo seleccionamos los puntos
# estimados generalizables a la región.

```

## Poblacionales: Cálculo de porcentajes (porcentajes de respuesta para la región)

- A la izquierda tenemos la secuencia completa para poder obtener resultados parciales de la figura 5. En particular, los resultados al ítem **MS3MR1**.
- Primero necesitamos **cargar los datos** de Cuestionario de Sexto Grado (**erce::erce\_2019\_qa6**), y los datos de cuestionario del módulo de habilidades socioemocionales (**erce::erce\_2019\_hse**).
- Luego de cargados los datos, necesitamos **ver el contenido** de los datos. En particular, queremos inspeccionar los datos del módulo de habilidades socioemocionales, porque queremos recuperar información específica de esta base de datos. Empleamos la función **dplyr::glimpse** para este cometido, luego de haber removido la meta data con **erce::remove\_labels**.
- Lo que queremos es **seleccionar** los datos específicos que queremos, los cuales son los datos de clustering, y las respuestas a los ítems. Aquí empleamos la función **dplyr::select()** para cumplir este propósito.
- Como hemos seleccionado de forma específica a las variables que queremos unir al cuestionario de estudiantes, nos estamos asegurando de no generar variables repetidas luego de la unión de bases de datos. Para unir los datos empleamos la función **dplyr::left\_join**, la cual nos permite agregar a la base de datos de Estudiantes de Sexto grado, las respuestas de cuestionario de Habilidades Socioemocionales.
- En la unión de base de datos, es importante que empleemos las variables de clustering completas. En este caso, estamos empleando la siguiente línea de código: **by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')**.
- Con los datos unidos, creamos una base de datos con **Taylor Series Linearization** (ver Heeringa et al., 2009), empleando a la librería **srvyr**. Empleamos este método de cálculo de errores, porque es más rápido que BRR para producir resultados, y produce resultados muy similares a los que genera BRR.
- Finalmente, empleamos la librería **srvyr** para producir los descriptivos de interés.

```

# figura 5
#
# abrir datos
#---
erce_qa6 <- erce::erce_2019_qa6 %>%
  erce::remove_labels()
erce_hse <- erce::erce_2019_hse %>%
  erce::remove_labels()
# inspeccionar datos-hse
#---
dplyr::glimpse(erce_hse)
#---
# seleccionar datos
#---
erce_hse_selected <- erce_hse %>%
  dplyr::select(
    # clustering
    IDCNTRY, ID SCHOOL, IDCLASS, IDSTUD,
    # items
    MS3MR1:VD3A06
  )
# unir datos y crear cluster únicos
#---
erce_a6 <- erce_qa6 %>%
  dplyr::left_join(., erce_hse_selected,
  by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')) %>%
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, "_", STRATA)))) %>%
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, "_", IDSCHOOL)))) %>%
  mutate(id_i = seq(1:nrow(.)))
# base de datos con diseño
#---
# survey method: taylor series linearization library(srvyr)
library(srvyr)
erce_tsl <- erce_a6 %>%
  as_survey_design(
    strata = id_s,
    ids = id_i,
    weights = ws,
    nest = TRUE)
# Opción: corrección a unidad primaria de muestreo que resulte
# única al estrato
library(survey)
options(survey.lonely.psu="adjust")
#---
# ítems de interés
#---
# MS3MR1
# Sigo las normas y reglas de la clase.
# MS3MR2
# Pido ayuda al profesor cuando no entiendo lo que hay que hacer.
# MS3PP2
# Aunque una tarea sea muy difícil, sigo trabajando en ella.
#---
# tabla de porcentajes para un ítem
#---
erce_tsl %>%
summarize(
  p1 = survey_mean(MS3MR1==1, na.rm = TRUE),
  p2 = survey_mean(MS3MR1==2, na.rm = TRUE),
  p3 = survey_mean(MS3MR1==3, na.rm = TRUE),
  p4 = survey_mean(MS3MR1==4, na.rm = TRUE),
  s1 = unweighted(mean(MS3MR1==1, na.rm = TRUE)),
  s2 = unweighted(mean(MS3MR1==2, na.rm = TRUE)),
  s3 = unweighted(mean(MS3MR1==3, na.rm = TRUE)),
  s4 = unweighted(mean(MS3MR1==4, na.rm = TRUE))
) %>%
mutate(item = 'MS3MR1') %>%
dplyr::select(item, p1, p2, p3, p4) %>%
knitr::kable(., digits = 2)

# Nota: aunque calculamos errores estandar, y
# puntos estimados sin diseño, por motivos
# ilustrativos, solo seleccionamos los puntos
# estimados generalizables a la región.

```

## Poblacionales: Cálculo de porcentajes (porcentajes de respuesta para la región)

- A la izquierda tenemos la secuencia completa para poder obtener resultados parciales de la figura 5. En particular, los resultados al ítem **MS3MR1**.
- Primero necesitamos **cargar los datos** de Cuestionario de Sexto Grado (**erce::erce\_2019\_qa6**), y los datos de cuestionario del módulo de habilidades socioemocionales (**erce::erce\_2019\_hse**).
- Luego de cargados los datos, necesitamos **ver el contenido** de los datos. En particular, queremos inspeccionar los datos del módulo de habilidades socioemocionales, porque queremos recuperar información específica de esta base de datos. Empleamos la función **dplyr::glimpse** para este cometido, luego de haber removido la meta data con **erce::remove\_labels**.
- Lo que queremos es **seleccionar** los datos específicos que queremos, los cuales son los datos de clustering, y las respuestas a los ítems. Aquí empleamos la función **dplyr::select()** para cumplir este propósito.
- Como hemos seleccionado de forma específica a las variables que queremos unir al cuestionario de estudiantes, nos estamos asegurando de no generar variables repetidas luego de la unión de bases de datos. Para unir los datos empleamos la función **dplyr::left\_join**, la cual nos permite agregar a la base de datos de Estudiantes de Sexto grado, las respuestas de cuestionario de Habilidades Socioemocionales.
- En la unión de base de datos, es importante que empleemos las variables de clustering completas. En este caso, estamos empleando la siguiente línea de código: **by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')**.
- Con los datos unidos, creamos una base de datos con **Taylor Series Linearization** (ver Heeringa et al., 2009), empleando a la librería **srvyr**. Empleamos este método de cálculo de errores, porque es más rápido que BRR para producir resultados, y produce resultados muy similares a los que genera BRR.
- Finalmente, empleamos la librería **srvyr** para producir los descriptivos de interés.

Ahora, revisemos esta secuencia por partes.

Problemas y Soluciones

# Abrir más de una base de datos y unirlas

Reproducido resultados de porcentajes (UNESCO-OREALC, 2021a, Figura 5)

## Abrir datos y Unir datos para producir resultados de HSE

```
# -----  
# figura 5  
# -----  
  
#-----  
# abrir datos  
#-----  
  
erce_qa6 <- erce::erce_2019_qa6 %>%  
  erce::remove_labels()  
  
erce_hse <- erce::erce_2019_hse %>%  
  erce::remove_labels()  
  
#-----  
# inspecccionar datos hse  
#-----  
  
names(erce_hse)  
  
[1] "IDSTUD"      "IDCLASS"      "IDSCHOOL"     "IDCNTRY"      "COUNTRY"  
[6] "STRATA"       "HSE"          "WT"           "WS"           "GRADE"  
[11] "MS3MR1"       "MS3MR2"       "MS3MR3"       "MS3MR4"       "MS3MR5"  
[16] "MS3MR6"       "MS3MR7"       "MS3PP1"       "MS3PP2"       "MS3PP3"  
[21] "MS3PP4"       "MS3PP5"       "MS3PP6"       "MS3PP7"       "MS3RE1"  
[26] "MS3RE2"       "MS3RE3"       "MS3RE4"       "MS3RE5"       "MS3RE6"  
[31] "E3TP1"        "E3TP2"        "E3TP3"        "E3TP4"        "E3TP5"  
[36] "E3DS1"        "E3DS2"        "E3DS3"        "E3DS4"        "E3DS5"  
[41] "VD3AV1"       "VD3AV2"       "VD3AV3"       "VD3AV4"       "VD3A01"  
[46] "VD3A02"       "VD3A03"       "VD3A04"       "VD3A05"       "VD3A06"  
[51] "MI6IT04_01"   "MI6IT04_02"  
  
#-----  
# seleccionar datos  
#-----  
  
erce_hse_selected <- erce_hse %>%  
  dplyr::select(  
    # clustering  
    IDCNTRY, IDSCHOOL, IDCLASS, IDSTUD,  
    # items  
    MS3MR1:VD3A06  
  )  
  
#-----  
# unir datos  
#-----  
  
erce_a6 <- erce_qa6 %>%  
  dplyr::left_join(., erce_hse_selected,  
    by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD'))
```

- En el código de la izquierda, incluimos una serie de pasos, desde la apertura de datos, hasta la unión de dos bases de datos. Nuestro objetivo con esta secuencia es crear una **base de datos única**, o **merge** de datos, la cual contenga la información de diseño del estudio, y las variables que queremos analizar.

## Abrir datos y Unir datos para producir resultados de HSE

```
# -----  
# figura 5  
# -----  
  
#-----  
# abrir datos  
#-----  
  
erce_qa6 <- erce::erce_2019_qa6 %>%  
  erce::remove_labels()  
  
erce_hse <- erce::erce_2019_hse %>%  
  erce::remove_labels()  
  
#-----  
# inspeccionar datos hse  
#-----  
  
names(erce_hse)
```

```
[1] "IDSTUD"      "IDCLASS"      "IDSCHOOL"     "IDCNTRY"      "COUNTRY"  
[6] "STRATA"       "HSE"          "WT"           "WS"           "GRADE"  
[11] "MS3MR1"       "MS3MR2"       "MS3MR3"       "MS3MR4"       "MS3MR5"  
[16] "MS3MR6"       "MS3MR7"       "MS3PP1"       "MS3PP2"       "MS3PP3"  
[21] "MS3PP4"       "MS3PP5"       "MS3PP6"       "MS3PP7"       "MS3RE1"  
[26] "MS3RE2"       "MS3RE3"       "MS3RE4"       "MS3RE5"       "MS3RE6"  
[31] "E3TP1"        "E3TP2"        "E3TP3"        "E3TP4"        "E3TP5"  
[36] "E3DS1"        "E3DS2"        "E3DS3"        "E3DS4"        "E3DS5"  
[41] "VD3AV1"       "VD3AV2"       "VD3AV3"       "VD3AV4"       "VD3A01"  
[46] "VD3A02"       "VD3A03"       "VD3A04"       "VD3A05"       "VD3A06"  
[51] "MI6IT04_01"   "MI6IT04_02"
```

```
#-----  
# seleccionar datos  
#-----  
  
erce_hse_selected <- erce_hse %>%  
  dplyr::select(  
    # clustering  
    IDCNTRY, IDSCHOOL, IDCLASS, IDSTUD,  
    # items  
    MS3MR1:VD3A06  
  )  
  
#-----  
# unir datos  
#-----  
  
erce_a6 <- erce_qa6 %>%  
  dplyr::left_join(., erce_hse_selected,  
    by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD'))
```

- En el código de la izquierda, incluimos una serie de pasos, desde la apertura de datos, hasta la unión de dos bases de datos. Nuestro objetivo con esta secuencia es crear una **base de datos única**, o **merge** de datos, la cual contenga la información de diseño del estudio, y las variables que queremos analizar.
- Cargamos los datos en la sección, empleando las líneas `erce:::erce_2019_qa6` y `erce:::erce_2019_hse`, y asignando cada una de estas a los objetos `erce_qa6` y `erce_hse` respectivamente. Además, removemos los *labels*, para facilitar el trabajo con las bases de datos.

## Abrir datos y Unir datos para producir resultados de HSE

```
# -----  
# figura 5  
# -----  
  
#-----  
# abrir datos  
#-----  
  
erce_qa6 <- erce::erce_2019_qa6 %>%  
  erce::remove_labels()  
  
erce_hse <- erce::erce_2019_hse %>%  
  erce::remove_labels()  
  
#-----  
# inspeccionar datos hse  
#-----  
  
names(erce_hse)  
  
[1] "IDSTUD"      "IDCLASS"      "IDSCHOOL"     "IDCNTRY"      "COUNTRY"  
[6] "STRATA"       "HSE"          "WT"           "WS"           "GRADE"  
[11] "MS3MR1"       "MS3MR2"       "MS3MR3"       "MS3MR4"       "MS3MR5"  
[16] "MS3MR6"       "MS3MR7"       "MS3PP1"       "MS3PP2"       "MS3PP3"  
[21] "MS3PP4"       "MS3PP5"       "MS3PP6"       "MS3PP7"       "MS3RE1"  
[26] "MS3RE2"       "MS3RE3"       "MS3RE4"       "MS3RE5"       "MS3RE6"  
[31] "E3TP1"        "E3TP2"        "E3TP3"        "E3TP4"        "E3TP5"  
[36] "E3DS1"        "E3DS2"        "E3DS3"        "E3DS4"        "E3DS5"  
[41] "VD3AV1"       "VD3AV2"       "VD3AV3"       "VD3AV4"       "VD3A01"  
[46] "VD3A02"       "VD3A03"       "VD3A04"       "VD3A05"       "VD3A06"  
[51] "MI6IT04_01"   "MI6IT04_02"  
  
#-----  
# seleccionar datos  
#-----  
  
erce_hse_selected <- erce_hse %>%  
  dplyr::select(  
    # clustering  
    IDCNTRY, IDSCHOOL, IDCLASS, IDSTUD,  
    # items  
    MS3MR1:VD3A06  
  )  
  
#-----  
# unir datos  
#-----  
  
erce_a6 <- erce_qa6 %>%  
  dplyr::left_join(., erce_hse_selected,  
    by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD'))
```

- En el código de la izquierda, incluimos una serie de pasos, desde la apertura de datos, hasta la unión de dos bases de datos. Nuestro objetivo con esta secuencia es crear una **base de datos única**, o **merge** de datos, la cual contenga la información de diseño del estudio, y las variables que queremos analizar.
- Cargamos los datos en la sección, empleando las líneas `erce:::erce_2019_qa6` y `erce:::erce_2019_hse`, y asignando cada una de estas a los objetos `erce_qa6` y `erce_hse` respectivamente. Además, removemos los *labels*, para facilitar el trabajo con las bases de datos.
- Para **inspeccionar** el contenido de `erce_hse`, en esta variante del código, empleamos la función `names()`, así podremos ver en la slide presente los nombres de las variables que nos interesan. Queremos identificar a las variables de clustering, y a las variables que alojan las respuestas a los ítems.

## Abrir datos y Unir datos para producir resultados de HSE

```
# -----  
# figura 5  
# -----  
  
#-----  
# abrir datos  
#-----  
  
erce_qa6 <- erce::erce_2019_qa6 %>%  
  erce::remove_labels()  
  
erce_hse <- erce::erce_2019_hse %>%  
  erce::remove_labels()  
  
#-----  
# inspeccionar datos hse  
#-----  
  
names(erce_hse)  
  
[1] "IDSTUD"      "IDCLASS"      "IDSCHOOL"     "IDCNTRY"      "COUNTRY"  
[6] "STRATA"       "HSE"          "WT"           "WS"           "GRADE"  
[11] "MS3MR1"       "MS3MR2"       "MS3MR3"       "MS3MR4"       "MS3MR5"  
[16] "MS3MR6"       "MS3MR7"       "MS3PP1"       "MS3PP2"       "MS3PP3"  
[21] "MS3PP4"       "MS3PP5"       "MS3PP6"       "MS3PP7"       "MS3RE1"  
[26] "MS3RE2"       "MS3RE3"       "MS3RE4"       "MS3RE5"       "MS3RE6"  
[31] "E3TP1"        "E3TP2"        "E3TP3"        "E3TP4"        "E3TP5"  
[36] "E3DS1"        "E3DS2"        "E3DS3"        "E3DS4"        "E3DS5"  
[41] "VD3AV1"       "VD3AV2"       "VD3AV3"       "VD3AV4"       "VD3A01"  
[46] "VD3A02"       "VD3A03"       "VD3A04"       "VD3A05"       "VD3A06"  
[51] "MI6IT04_01"   "MI6IT04_02"  
  
#-----  
# seleccionar datos  
#-----  
  
erce_hse_selected <- erce_hse %>%  
  dplyr::select(  
    # clustering  
    IDCNTRY, IDSCHOOL, IDCLASS, IDSTUD,  
    # items  
    MS3MR1:VD3A06  
  )  
  
#-----  
# unir datos  
#-----  
  
erce_a6 <- erce_qa6 %>%  
  dplyr::left_join(., erce_hse_selected,  
    by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD'))
```

- En el código de la izquierda, incluimos una serie de pasos, desde la apertura de datos, hasta la unión de dos bases de datos. Nuestro objetivo con esta secuencia es crear una **base de datos única**, o **merge** de datos, la cual contenga la información de diseño del estudio, y las variables que queremos analizar.
- Cargamos los datos en la sección, empleando las líneas `erce:::erce_2019_qa6` y `erce:::erce_2019_hse`, y asignando cada una de estas a los objetos `erce_qa6` y `erce_hse` respectivamente. Además, removemos los *labels*, para facilitar el trabajo con las bases de datos.
- Para **inspeccionar** el contenido de `erce_hse`, en esta variante del código, empleamos la función `names()`, así podremos ver en la slide presente los nombres de las variables que nos interesan. Queremos identificar a las variables de clustering, y a las variables que alojan las respuestas a los ítems.
- Luego, creamos un objeto solo con las variables seleccionadas empleando el comando `dplyr::select()`. Este comando nos permite nombrar las variables que queremos seleccionar, y tambien nos permite seleccionar a todas las variables que se encuentren en un rango. Lo anterior quiere decir que, si indicamos `MS3MR1:VD3A06` al interior de `dplyr::select()`, se incluyen a todas las variables que se encuentran entre **MS3MR1** y **VD3AO6**.

## Abrir datos y Unir datos para producir resultados de HSE

```
# -----  
# figura 5  
# -----  
  
#-----  
# abrir datos  
#-----  
  
erce_qa6 <- erce::erce_2019_qa6 %>%  
  erce::remove_labels()  
  
erce_hse <- erce::erce_2019_hse %>%  
  erce::remove_labels()  
  
#-----  
# inspeccionar datos hse  
#-----  
  
names(erce_hse)  
  
[1] "IDSTUD"      "IDCLASS"      "IDSCHOOL"     "IDCNTRY"      "COUNTRY"  
[6] "STRATA"       "HSE"          "WT"           "WS"           "GRADE"  
[11] "MS3MR1"       "MS3MR2"       "MS3MR3"       "MS3MR4"       "MS3MR5"  
[16] "MS3MR6"       "MS3MR7"       "MS3PP1"       "MS3PP2"       "MS3PP3"  
[21] "MS3PP4"       "MS3PP5"       "MS3PP6"       "MS3PP7"       "MS3RE1"  
[26] "MS3RE2"       "MS3RE3"       "MS3RE4"       "MS3RE5"       "MS3RE6"  
[31] "E3TP1"        "E3TP2"        "E3TP3"        "E3TP4"        "E3TP5"  
[36] "E3DS1"        "E3DS2"        "E3DS3"        "E3DS4"        "E3DS5"  
[41] "VD3AV1"       "VD3AV2"       "VD3AV3"       "VD3AV4"       "VD3A01"  
[46] "VD3A02"       "VD3A03"       "VD3A04"       "VD3A05"       "VD3A06"  
[51] "MI6IT04_01"   "MI6IT04_02"  
  
#-----  
# seleccionar datos  
#-----  
  
erce_hse_selected <- erce_hse %>%  
  dplyr::select(  
    # clustering  
    IDCNTRY, IDSCHOOL, IDCLASS, IDSTUD,  
    # items  
    MS3MR1:VD3A06  
  )  
  
#-----  
# unir datos  
#-----  
  
erce_a6 <- erce_qa6 %>%  
  dplyr::left_join(., erce_hse_selected,  
    by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD'))
```

- En el código de la izquierda, incluimos una serie de pasos, desde la apertura de datos, hasta la unión de dos bases de datos. Nuestro objetivo con esta secuencia es crear una **base de datos única**, o **merge** de datos, la cual contenga la información de diseño del estudio, y las variables que queremos analizar.
- Cargamos los datos en la sección, empleando las líneas `erce:::erce_2019_qa6` y `erce:::erce_2019_hse`, y asignando cada una de estas a los objetos `erce_qa6` y `erce_hse` respectivamente. Además, removemos los *labels*, para facilitar el trabajo con las bases de datos.
- Para **inspeccionar** el contenido de `erce_hse`, en esta variante del código, empleamos la función `names()`, así podremos ver en la slide presente los nombres de las variables que nos interesan. Queremos identificar a las variables de clustering, y a las variables que alojan las respuestas a los ítems.
- Luego, creamos un objeto solo con las variables seleccionadas empleando el comando `dplyr::select()`. Este comando nos permite nombrar las variables que queremos seleccionar, y tambien nos permite seleccionar a todas las variables que se encuentren en un rango. Lo anterior quiere decir que, si indicamos `MS3MR1:VD3A06` al interior de `dplyr::select()`, se incluyen a todas las variables que se encuentran entre **MS3MR1** y **VD3AO6**.
- Finalmente, **unimos** los datos empleando a `dplyr::left_join`. Esta función emplea a la primera base de datos como base de datos clave. Es decir, que estamos empleando `erce_qa6` como tabla fija, sobre la cual agregaremos toda la información que se pueda obtener de la base de datos `erce_hse`. De manera tal, que si hubiera un caso que posee datos en `erce_hse`, pero no posee datos en `erce_qa6`, entonces este caso sería descartado de la unión.

Problemas y Soluciones

# Generación de variables de diseño

Reproducido resultados regionales (UNESCO-OREALC, 2021a, Figura 5)

## Generación de **variables de diseño** datos para producir resultados de HSE

```
# -----  
# figura 5  
# -----  
# [...] continua código anterior]  
#-----  
# unir datos y crear cluster únicos  
#-----  
  
erce_a6 <- erce_qa6 %>%  
  dplyr::left_join(., erce_hse_selected,  
    by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')) %>%  
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%  
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, "_", STRATA)))) %>%  
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, "_", IDSCHOOL)))) %>%  
  mutate(id_i = seq(1:nrow(.)))  
#-----  
# base de datos con diseño  
#-----  
  
# survey method: taylor series linearization library(srvyr)  
library(srvyr)  
erce_tsl <- erce_a6 %>%  
  as_survey_design(  
    strata = id_s,  
    ids = id_j,  
    weights = ws,  
    nest = TRUE)  
  
# Opción: corrección a unidad primaria de muestreo que resulte  
# única al estrato  
  
library(survey)  
options(survey.lonely.psu="adjust")
```

**Nota:** empleamos **TSL** porque para producir los resultados de la figura 5, **TSL** es más rápido para generar los resultados con **BRR**.

- Para la unión de datos, estamos aprovechando una propiedad de la función **dplyr::left\_join**. Si en el argumento **by** = incluimos en una lista el conjunto de variables que hacen que una fila sea única en la base de datos, esta lista será suficiente para unir ambas tablas de datos, sin repetir casos. Esta lista de atributos, que hace que un caso sea único se conforma de todos las variables siguientes: **c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')**.

## Generación de **variables de diseño** datos para producir resultados de HSE

```
# -----  
# figura 5  
# -----  
  
# [...] continua código anterior]  
  
#-----  
# unir datos y crear cluster únicos  
#-----  
  
erce_a6 <- erce_qa6 %>%  
  dplyr::left_join(., erce_hse_selected,  
    by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')) %>%  
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%  
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, "_", STRATA)))) %>%  
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, "_", IDSCHOOL)))) %>%  
  mutate(id_i = seq(1:nrow(.)))  
  
#-----  
# base de datos con diseño  
#-----  
  
# survey method: taylor series linearization library(srvyr)  
library(srvyr)  
erce_tsl <- erce_a6 %>%  
  as_survey_design(  
    strata = id_s,  
    ids = id_j,  
    weights = ws,  
    nest = TRUE)  
  
# Opción: corrección a unidad primaria de muestreo que resulte  
# única al estrato  
  
library(survey)  
options(survey.lonely.psu="adjust")
```

**Nota:** empleamos **TSL** porque para producir los resultados de la figura 5, **TSL** es más rápido para generar los resultados con **BRR**.

- Para la unión de datos, estamos aprovechando una propiedad de la función **dplyr::left\_join**. Si en el argumento **by** = incluimos en una lista el conjunto de variables que hacen que una fila sea única en las base de datos, esta lista será suficiente para unir ambas tablas de datos, sin repetir casos. Esta lista de atributos, que hace que un caso sea único se conforma de todos las variables siguientes: **c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')**.
- Lo anterior es necesario, porque el identificador de las escuelas es único al interior del país; pero no único entre países. En consecuencia, en los datos de cada país existe la escuela **1001**. No obstante, mientras incluyamos en la lista crítica de variables en el argumento **by** =, la unión de datos será exitosa.

## Generación de **variables de diseño** datos para producir resultados de HSE

```
# -----  
# figura 5  
# -----  
  
# [...] continua código anterior]  
  
#-----  
# unir datos y crear cluster únicos  
#-----  
  
erce_a6 <- erce_qa6 %>%  
  dplyr::left_join(., erce_hse_selected,  
    by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')) %>%  
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%  
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, "_", STRATA)))) %>%  
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, "_", IDSCHOOL)))) %>%  
  mutate(id_i = seq(1:nrow(.)))  
  
#-----  
# base de datos con diseño  
#-----  
  
# survey method: taylor series linearization library(srvyr)  
library(srvyr)  
erce_tsl <- erce_a6 %>%  
  as_survey_design(  
    strata = id_s,  
    ids = id_j,  
    weights = ws,  
    nest = TRUE)  
  
# Opción: corrección a unidad primaria de muestreo que resulte  
# única al estrato  
  
library(survey)  
options(survey.lonely.psu="adjust")
```

**Nota:** empleamos **TSL** porque para producir los resultados de la figura 5, **TSL** es más rápido para generar los resultados con **BRR**.

- Para la unión de datos, estamos aprovechando una propiedad de la función **dplyr::left\_join**. Si en el argumento **by** = incluimos en una lista el conjunto de variables que hacen que una fila sea única en las base de datos, esta lista será suficiente para unir ambas tablas de datos, sin repetir casos. Esta lista de atributos, que hace que un caso sea único se conforma de todos las variables siguientes: **c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')**.
- Lo anterior es necesario, porque el identificador de las escuelas es único al interior del país; pero no único entre países. En consecuencia, en los datos de cada país existe la escuela **1001**. No obstante, mientras incluyamos en la lista crítica de variables en el argumento **by** =, la unión de datos será exitosa.
- Luego, necesitamos variables de diseño o variables de **clustering** que sean únicas. De esta manera, creamos a las variables **id\_k, id\_s, id\_j, id\_i**, para tener un vector numérico, que nos permita identificar a países, estratos, escuelas, y estudiantes con un identificador único entre países. Debido a que estamos interesados en generar un **resultado poblacional a la región**, es necesario que tengamos variables de clustering que sean únicas entre los países.

## Generación de **variables de diseño** datos para producir resultados de HSE

```
# -----  
# figura 5  
# -----  
  
# [...] continua código anterior]  
  
#-----  
# unir datos y crear cluster únicos  
#-----  
  
erce_a6 <- erce_qa6 %>%  
  dplyr::left_join(., erce_hse_selected,  
    by = c('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')) %>%  
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%  
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, "_", STRATA)))) %>%  
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, "_", IDSCHOOL)))) %>%  
  mutate(id_i = seq(1:nrow(.)))  
  
#-----  
# base de datos con diseño  
#-----  
  
# survey method: taylor series linearization library(srvyr)  
library(srvyr)  
erce_tsl <- erce_a6 %>%  
  as_survey_design(  
    strata = id_s,  
    ids = id_j,  
    weights = ws,  
    nest = TRUE)  
  
# Opción: corrección a unidad primaria de muestreo que resulte  
# única al estrato  
  
library(survey)  
options(survey.lonely.psu="adjust")
```

**Nota:** empleamos **TSL** porque para producir los resultados de la figura 5, **TSL** es más rápido para generar los resultados con **BRR**.

- Para la unión de datos, estamos aprovechando una propiedad de la función **dplyr::left\_join**. Si en el argumento **by** = incluimos en una lista el conjunto de variables que hacen que una fila sea única en las base de datos, esta lista será suficiente para unir ambas tablas de datos, sin repetir casos. Esta lista de atributos, que hace que un caso sea único se conforma de todos las variables siguientes: **c ('IDCNTRY', 'IDSCHOOL', 'IDCLASS', 'IDSTUD')**.
- Lo anterior es necesario, porque el identificador de las escuelas es único al interior del país; pero no único entre países. En consecuencia, en los datos de cada país existe la escuela **1001**. No obstante, mientras incluyamos en la lista crítica de variables en el argumento **by** =, la unión de datos será exitosa.
- Luego, necesitamos variables de diseño o variables de **clustering** que sean únicas. De esta manera, creamos a las variables **id\_k, id\_s, id\_j, id\_i**, para tener un vector numérico, que nos permita identificar a países, estratos, escuelas, y estudiantes con un identificador único entre países. Debido a que estamos interesados en generar un **resultado poblacional a la región**, es necesario que tengamos variables de clustering que sean únicas entre los países.
- Teniendo a todos los componentes anteriores, las variables de clustering únicas, y las variables de respuesta de interés, podemos crear un objeto que especifique el diseño del estudio. En este caso emplearemos Taylor Series Linearization (**TSL**), de la librería **srvyr**. Este método de estimación de errores nos entregara resultados similares a los **BRR**. En la literatura de estudios de gran escala, se tiende a favorecer a BRR, porque produciría errores de mejor calidad para la estimación de cuartiles (Rust et al, 2017), y mejor cobertura en la recuperación de parámetros con intervalos de confianza (Heeringa et al. 2009). No obstante, para generar puntos estimados TSL es más eficiente.

Problemas y Soluciones

# Cálculo de porcentajes regionales

Reproducido resultados regionales (UNESCO-OREALC, 2021a, Figura 5)

## Cálculo de porcentajes de respuesta por ítem para reproducir resultados de HSE

```
# -----  
# figura 5  
# -----  
# [...] continua código anterior]  
#-----  
# items de interés  
#-----  
  
# MS3MR1 Sigo las normas y reglas de la clase.  
# MS3MR2 Pido ayuda al profesor cuando no entiendo lo que hay que hacer.  
# MS3PP2 Aunque una tarea sea muy difícil, sigo trabajando en ella.  
  
#-----  
# porcentajes por ítem  
#-----  
  
row_1 <- erce tsl %>%  
  summarize(  
    p1 = survey_mean(MS3MR1==1, na.rm = TRUE),  
    p2 = survey_mean(MS3MR1==2, na.rm = TRUE),  
    p3 = survey_mean(MS3MR1==3, na.rm = TRUE),  
    p4 = survey_mean(MS3MR1==4, na.rm = TRUE),  
    ) %>%  
  mutate(item = 'MS3MR1') %>%  
  dplyr::select(item, p1, p2, p3, p4)  
  
row_2 <- erce tsl %>%  
  summarize(  
    p1 = survey_mean(MS3MR2==1, na.rm = TRUE),  
    p2 = survey_mean(MS3MR2==2, na.rm = TRUE),  
    p3 = survey_mean(MS3MR2==3, na.rm = TRUE),  
    p4 = survey_mean(MS3MR2==4, na.rm = TRUE),  
    ) %>%  
  mutate(item = 'MS3MR2') %>%  
  dplyr::select(item, p1, p2, p3, p4)  
  
row_3 <- erce tsl %>%  
  summarize(  
    p1 = survey_mean(MS3PP2==1, na.rm = TRUE),  
    p2 = survey_mean(MS3PP2==2, na.rm = TRUE),  
    p3 = survey_mean(MS3PP2==3, na.rm = TRUE),  
    p4 = survey_mean(MS3PP2==4, na.rm = TRUE),  
    ) %>%  
  mutate(item = 'MS3PP2') %>%  
  dplyr::select(item, p1, p2, p3, p4)  
  
#-----  
# tabla  
#-----  
  
tabla_fig5 <- dplyr::bind_rows(row_1, row_2, row_3)  
#-----  
# mostrar tabla  
#-----  
  
knitr::kable(tabla_fig5, digits = 2)
```

- Empleando a la librería **svyrr**, y sus funciones de descriptivos, mezclados con las función de **summarize**, es sencillo producir los resultados de interés por ítem. Una vez **unidos** los objetos **row\_1**, **row\_2** y **row\_3**, podemos generar los resultados que buscamos con la línea **knitr::kable(tabla\_fig5, digits = 2)**.

## Cálculo de porcentajes de respuesta por ítem para reproducir resultados de HSE

```
# -----  
# figura 5  
# -----  
# [...] continua código anterior]  
#-----  
# items de interés  
#-----  
  
# MS3MR1 Sigo las normas y reglas de la clase.  
# MS3MR2 Pido ayuda al profesor cuando no entiendo lo que hay que hacer.  
# MS3PP2 Aunque una tarea sea muy difícil, sigo trabajando en ella.  
  
#-----  
# porcentajes por ítem  
#-----  
  
row_1 <- erce tsl %>%  
  summarize(  
    p1 = survey_mean(MS3MR1==1, na.rm = TRUE),  
    p2 = survey_mean(MS3MR1==2, na.rm = TRUE),  
    p3 = survey_mean(MS3MR1==3, na.rm = TRUE),  
    p4 = survey_mean(MS3MR1==4, na.rm = TRUE),  
    ) %>%  
  mutate(item = 'MS3MR1') %>%  
  dplyr::select(item, p1, p2, p3, p4)  
  
row_2 <- erce tsl %>%  
  summarize(  
    p1 = survey_mean(MS3MR2==1, na.rm = TRUE),  
    p2 = survey_mean(MS3MR2==2, na.rm = TRUE),  
    p3 = survey_mean(MS3MR2==3, na.rm = TRUE),  
    p4 = survey_mean(MS3MR2==4, na.rm = TRUE),  
    ) %>%  
  mutate(item = 'MS3MR2') %>%  
  dplyr::select(item, p1, p2, p3, p4)  
  
row_3 <- erce tsl %>%  
  summarize(  
    p1 = survey_mean(MS3PP2==1, na.rm = TRUE),  
    p2 = survey_mean(MS3PP2==2, na.rm = TRUE),  
    p3 = survey_mean(MS3PP2==3, na.rm = TRUE),  
    p4 = survey_mean(MS3PP2==4, na.rm = TRUE),  
    ) %>%  
  mutate(item = 'MS3PP2') %>%  
  dplyr::select(item, p1, p2, p3, p4)  
  
#-----  
# tabla  
#-----  
  
tabla_fig5 <- dplyr::bind_rows(row_1, row_2, row_3)  
#-----  
# mostrar tabla  
#-----  
  
knitr::kable(tabla_fig5, digits = 2)
```

- Empleando a la librería **svyrr**, y sus funciones de descriptivos, mezclados con las función de **summarize**, es sencillo producir los resultados de interés por ítem. Una vez **unidos** los objetos **row\_1**, **row\_2** y **row\_3**, podemos generar los resultados que buscamos con la línea **knitr::kable(tabla\_fig5, digits = 2)**.

# Resultados graficados en la Figura 5 (UNESCO-OREALC, 2021a)

item	p1	p2	p3	p4
MS3MR1	0.03	0.17	0.26	0.54
MS3MR2	0.05	0.30	0.30	0.35
MS3PP2	0.04	0.18	0.28	0.51

## Cálculo de porcentajes de respuesta por ítem para reproducir resultados de HSE

```
# -----  
# figura 5  
# -----  
# [...] continua código anterior]  
#-----  
# items de interés  
#-----  
  
# MS3MR1 Sigo las normas y reglas de la clase.  
# MS3MR2 Pido ayuda al profesor cuando no entiendo lo que hay que hacer.  
# MS3PP2 Aunque una tarea sea muy difícil, sigo trabajando en ella.  
  
#-----  
# porcentajes por ítem  
#-----  
  
row_1 <- erce tsl %>%  
  summarize(  
    p1 = survey_mean(MS3MR1==1, na.rm = TRUE),  
    p2 = survey_mean(MS3MR1==2, na.rm = TRUE),  
    p3 = survey_mean(MS3MR1==3, na.rm = TRUE),  
    p4 = survey_mean(MS3MR1==4, na.rm = TRUE),  
  ) %>%  
  mutate(item = 'MS3MR1') %>%  
  dplyr::select(item, p1, p2, p3, p4)  
  
row_2 <- erce tsl %>%  
  summarize(  
    p1 = survey_mean(MS3MR2==1, na.rm = TRUE),  
    p2 = survey_mean(MS3MR2==2, na.rm = TRUE),  
    p3 = survey_mean(MS3MR2==3, na.rm = TRUE),  
    p4 = survey_mean(MS3MR2==4, na.rm = TRUE),  
  ) %>%  
  mutate(item = 'MS3MR2') %>%  
  dplyr::select(item, p1, p2, p3, p4)  
  
row_3 <- erce tsl %>%  
  summarize(  
    p1 = survey_mean(MS3PP2==1, na.rm = TRUE),  
    p2 = survey_mean(MS3PP2==2, na.rm = TRUE),  
    p3 = survey_mean(MS3PP2==3, na.rm = TRUE),  
    p4 = survey_mean(MS3PP2==4, na.rm = TRUE),  
  ) %>%  
  mutate(item = 'MS3PP2') %>%  
  dplyr::select(item, p1, p2, p3, p4)  
  
#-----  
# tabla  
#-----  
  
tabla_fig5 <- dplyr::bind_rows(row_1, row_2, row_3)  
#-----  
# mostrar tabla  
#-----  
  
knitr::kable(tabla_fig5, digits = 2)
```

- Empleando a la librería **srvyr**, y sus funciones de descriptivos, mezclados con las función de **summarize**, es sencillo producir los resultados de interés por ítem. Una vez **unidos** los objetos **row\_1**, **row\_2** y **row\_3**, podemos generar los resultados que buscamos con la línea **knitr::kable(tabla\_fig5, digits = 2)**.

# Resultados graficados en la Figura 5 (UNESCO-OREALC, 2021a)

item	p1	p2	p3	p4
MS3MR1	0.03	0.17	0.26	0.54
MS3MR2	0.05	0.30	0.30	0.35
MS3PP2	0.04	0.18	0.28	0.51

- El cálculo de porcentajes de respuestas categóricas, empleando diseño, tiende a ser más complejo que en el caso de cálculos de frecuencias y porcentajes de base de datos sin diseños. En este caso, estamos aprovechando una característica de la librería **srvyr**, que en su función **survey\_mean**, acepta como atajo, declarar **nombre\_variable == 1** y con esto generar un cálculo de porcentaje de repuesta al valor 1.

## Cálculo de porcentajes de respuesta por ítem para reproducir resultados de HSE

```
# -----  
# figura 5  
# -----  
# [...] continua código anterior]  
#-----  
# items de interés  
#-----  
  
# MS3MR1 Sigo las normas y reglas de la clase.  
# MS3MR2 Pido ayuda al profesor cuando no entiendo lo que hay que hacer.  
# MS3PP2 Aunque una tarea sea muy difícil, sigo trabajando en ella.  
  
#-----  
# porcentajes por ítem  
#-----  
  
row_1 <- erce tsl %>%  
  summarize(  
    p1 = survey_mean(MS3MR1==1, na.rm = TRUE),  
    p2 = survey_mean(MS3MR1==2, na.rm = TRUE),  
    p3 = survey_mean(MS3MR1==3, na.rm = TRUE),  
    p4 = survey_mean(MS3MR1==4, na.rm = TRUE),  
  ) %>%  
  mutate(item = 'MS3MR1') %>%  
  dplyr::select(item, p1, p2, p3, p4)  
  
row_2 <- erce tsl %>%  
  summarize(  
    p1 = survey_mean(MS3MR2==1, na.rm = TRUE),  
    p2 = survey_mean(MS3MR2==2, na.rm = TRUE),  
    p3 = survey_mean(MS3MR2==3, na.rm = TRUE),  
    p4 = survey_mean(MS3MR2==4, na.rm = TRUE),  
  ) %>%  
  mutate(item = 'MS3MR2') %>%  
  dplyr::select(item, p1, p2, p3, p4)  
  
row_3 <- erce tsl %>%  
  summarize(  
    p1 = survey_mean(MS3PP2==1, na.rm = TRUE),  
    p2 = survey_mean(MS3PP2==2, na.rm = TRUE),  
    p3 = survey_mean(MS3PP2==3, na.rm = TRUE),  
    p4 = survey_mean(MS3PP2==4, na.rm = TRUE),  
  ) %>%  
  mutate(item = 'MS3PP2') %>%  
  dplyr::select(item, p1, p2, p3, p4)  
  
#-----  
# tabla  
#-----  
  
tabla_fig5 <- dplyr::bind_rows(row_1, row_2, row_3)  
#-----  
# mostrar tabla  
#-----  
  
knitr::kable(tabla_fig5, digits = 2)
```

- Empleando a la librería **srvyr**, y sus funciones de descriptivos, mezclados con las función de **summarize**, es sencillo producir los resultados de interés por ítem. Una vez **unidos** los objetos **row\_1**, **row\_2** y **row\_3**, podemos generar los resultados que buscamos con la línea **knitr::kable(tabla\_fig5, digits = 2)**.

# Resultados graficados en la Figura 5 (UNESCO-OREALC, 2021a)

ítem	p1	p2	p3	p4
MS3MR1	0.03	0.17	0.26	0.54
MS3MR2	0.05	0.30	0.30	0.35
MS3PP2	0.04	0.18	0.28	0.51

- El cálculo de porcentajes de respuestas categóricas, empleando diseño, tiende a ser más complejo que en el caso de cálculos de frecuencias y porcentajes de base de datos sin diseños. En este caso, estamos aprovechando una característica de la librería **srvyr**, que en su función **survey\_mean**, acepta como atajo, declarar **nombre\_variable == 1** y con esto generar un cálculo de porcentaje de repuesta al valor 1.
- Con el atajo anteriormente mencionado, podemos proceder a obtener el porcentaje de respuesta de cada opción en cada ítem. De esta forma, nos basta con variar el argumento **== 1**, **== 2**, **== 3** y **== 4** de forma correspondiente y obtener los 12 puntos estimados de interés (3 ítems x 4 categorías de respuesta).

## Cálculo de porcentajes de respuesta por ítem para reproducir resultados de HSE

```
# -----  
# figura 5  
# -----  
# [...] continua código anterior]  
#-----  
# items de interés  
#-----  
  
# MS3MR1 Sigo las normas y reglas de la clase.  
# MS3MR2 Pido ayuda al profesor cuando no entiendo lo que hay que hacer.  
# MS3PP2 Aunque una tarea sea muy difícil, sigo trabajando en ella.  
  
#-----  
# porcentajes por ítem  
#-----  
  
row_1 <- erce tsl %>%  
  summarize(  
    p1 = survey_mean(MS3MR1==1, na.rm = TRUE),  
    p2 = survey_mean(MS3MR1==2, na.rm = TRUE),  
    p3 = survey_mean(MS3MR1==3, na.rm = TRUE),  
    p4 = survey_mean(MS3MR1==4, na.rm = TRUE),  
  ) %>%  
  mutate(item = 'MS3MR1') %>%  
  dplyr::select(item, p1, p2, p3, p4)  
  
row_2 <- erce tsl %>%  
  summarize(  
    p1 = survey_mean(MS3MR2==1, na.rm = TRUE),  
    p2 = survey_mean(MS3MR2==2, na.rm = TRUE),  
    p3 = survey_mean(MS3MR2==3, na.rm = TRUE),  
    p4 = survey_mean(MS3MR2==4, na.rm = TRUE),  
  ) %>%  
  mutate(item = 'MS3MR2') %>%  
  dplyr::select(item, p1, p2, p3, p4)  
  
row_3 <- erce tsl %>%  
  summarize(  
    p1 = survey_mean(MS3PP2==1, na.rm = TRUE),  
    p2 = survey_mean(MS3PP2==2, na.rm = TRUE),  
    p3 = survey_mean(MS3PP2==3, na.rm = TRUE),  
    p4 = survey_mean(MS3PP2==4, na.rm = TRUE),  
  ) %>%  
  mutate(item = 'MS3PP2') %>%  
  dplyr::select(item, p1, p2, p3, p4)  
  
#-----  
# tabla  
#-----  
  
tabla_fig5 <- dplyr::bind_rows(row_1, row_2, row_3)  
  
#-----  
# mostrar tabla  
#-----  
  
knitr::kable(tabla_fig5, digits = 2)
```

- Empleando a la librería **srvyr**, y sus funciones de descriptivos, mezclados con las función de **summarize**, es sencillo producir los resultados de interés por ítem. Una vez **unidos** los objetos **row\_1**, **row\_2** y **row\_3**, podemos generar los resultados que buscamos con la línea **knitr::kable(tabla\_fig5, digits = 2)**.

# Resultados graficados en la Figura 5 (UNESCO-OREALC, 2021a)

item	p1	p2	p3	p4
MS3MR1	0.03	0.17	0.26	0.54
MS3MR2	0.05	0.30	0.30	0.35
MS3PP2	0.04	0.18	0.28	0.51

- El cálculo de porcentajes de respuestas categóricas, empleando diseño, tiende a ser más complejo que en el caso de cálculos de frecuencias y porcentajes de base de datos sin diseños. En este caso, estamos aprovechando una característica de la librería **srvyr**, que en su función **survey\_mean**, acepta como atajo, declarar **nombre\_variable == 1** y con esto generar un cálculo de porcentaje de repuesta al valor 1.
- Con el atajo anteriormente mencionado, podemos proceder a obtener el porcentaje de respuesta de cada opción en cada ítem. De esta forma, nos basta con variar el argumento **== 1, == 2, == 3 y == 4** de forma correspondiente y obtener los 12 puntos estimados de interés (3 ítems x 4 categorías de respuesta).
- Una vez que tenemos los puntos estimados en los objetos **row\_1**, **row\_2** y **row\_3**, unimos cada una de estas filas de datos en una tabla, empleando la función **dplyr::bind\_rows**. Esta función une filas, o apila tablas *hacia abajo*. En nuestro caso, con esta función generamos una tabla donde **row\_1** es la primera fila, **row\_2** es la segunda fila, y **row\_3** es la tercera fila. Finalmente, a **tabla\_fig5** asignamos la tabla de interés, y con todo lo anterior ya tenemos los resultados presentados en la Figura 5.

Problemas y Soluciones

# Transferibilidad y Reproducibilidad

Compartir resultados generados y código reoproducible

# Transferibilidad y Reproducibilidad

```

224
225 # Items de interes
226
227
228 # NSM03 Sigue las normas y reglas de la clase
229 # NSM04 Hace lo que se le dice sin necesariamente lo que hay que hacer.
230 # NSM02 Aunque una tarea sea muy dificil, sigue trabajando en ella.
231
232 # porcentajes por tarea
233
234
235 row_1 <- erce_tst %>
236   p0 <- survey_mean(NSM01_1, na.rm = TRUE),
237   p1 <- survey_mean(NSM01_2, na.rm = TRUE),
238   p2 <- survey_mean(NSM01_3, na.rm = TRUE),
239   p3 <- survey_mean(NSM01_4, na.rm = TRUE),
240
241   mutate(item = "NSM01") %>
242     dplyr::select(tareas, p0, p1, p2, p3)
243
244 row_2 <- erce_tst %>
245   p0 <- survey_mean(NSM02_1, na.rm = TRUE),
246   p1 <- survey_mean(NSM02_2, na.rm = TRUE),
247   p2 <- survey_mean(NSM02_3, na.rm = TRUE),
248   p3 <- survey_mean(NSM02_4, na.rm = TRUE),
249
250   mutate(item = "NSM02") %>
251     dplyr::select(tareas, p0, p1, p2, p3)
252
253 row_3 <- erce_tst %>
254   p0 <- survey_mean(NSM02_1, na.rm = TRUE),
255   p1 <- survey_mean(NSM02_2, na.rm = TRUE),
256   p2 <- survey_mean(NSM02_3, na.rm = TRUE),
257   p3 <- survey_mean(NSM02_4, na.rm = TRUE),
258
259   mutate(item = "NSM02") %>
260     dplyr::select(tareas, p0, p1, p2, p3)
261
262
263
264
265
266
267
268 ## Definir forma de presentacion en resultados
269
270 ## echo = TRUE, eval = TRUE
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
999

```



- Todos los pasos revisados hasta ahora, los juntamos en un solo **código**. El código generado cubre diferentes pasos tales como:
    - la apertura de los datos
    - inspección de datos
    - unión de bases de datos diferentes
    - creación de variables de diseño comunes, para estimar resultados regionales
    - creación de base de datos para estimaciones con **TSL**
    - estimación de porcentajes de variables categóricas
    - creación de tablas uniendo objetos
    - exportar resultados a excel

**Nota:** imagen de cómo se ve el código reproducible generado en formato **RMarkdown**. A la izquierda se encuentra el código escrito en Sublime Text, mientras que a la derecha se encuentra un screenshot del output generado en html.

Este código se puede bajar desde el siguiente link:

[https://github.com/dacarras/erce\\_2022\\_lsa/blob/main/ejemplo\\_figura\\_5\\_hse.rmd](https://github.com/dacarras/erce_2022_lsa/blob/main/ejemplo_figura_5_hse.rmd)

```
# -----
# figura 5
#
#-----
# exportar resultados
#-----
```

```
tabla_fig5 %>%
openxlsx::write.xlsx(..,
  'tabla_figura_5.xlsx',
  overwrite = TRUE)
```

Poblacionales

# Cálculo de medias y cuartiles

Reproducido resultados de porcentajes (UNESCO-OREALC, 2021a, Figura 6)

## Cálculo de medias y cuartiles por país

```
# figura 6
# unir datos y crear cluster únicos
# función
#
erce_a6 <- erce_qa6 %>%
  erce::remove_labels() %>%
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, " ", STRATA)))) %>%
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, " ", ID SCHOOL)))) %>%
  mutate(id_i = seq(1:nrow(.)))

# diseño (BRR)
library(srvyr)
erce_brr <- data %>%
  erce::remove_labels() %>%
  as_survey_rep(type = ' Fay',
  repweights = starts_with('BRR'),
  weights = 'WT',
  combined_weights = TRUE,
  rho = .5,
  mse = TRUE)

ctry_acron <- data %>%
  dplyr::select(COUNTRY) %>%
  unique() %>%
  dplyr::pull()

# calcular tabla
loc_est <- erce_brr %>%
  summarize(
    e = survey_mean({{var}}, na.rm=TRUE, vartype = 'ci'),
    q = survey_quantile({{var}}, na.rm=TRUE, c(.10, .25, .75, .90)),
    sd = survey_sd({{var}}, na.rm = TRUE)
  ) %>%
  rename(l1 = e_low) %>%
  rename(u1 = e_up) %>%
  rename(p10 = q.q10) %>%
  rename(p25 = q.q25) %>%
  rename(p75 = q.q75) %>%
  rename(p90 = q.q90) %>%
  mutate(ctry = ctry_acron) %>%
  dplyr::select(ctry, e, l1, u1, p10, p25, p75, p90, sd
  )

return(loc_est)
}

# definir dígitos en consola
#
options(digits = 10)
options(scipen = 999999)

# resultado por país
#
dplyr::filter(erce_a6, COUNTRY == 'ARG') %>%
  location_est(data = ., var = AURES) %>%
  knitr::kable(., digits = 2)

|ctry |   e|   l1|   u1|   p10|   p25|   p75|   p90|   sd|
|----:|---:|---:|---:|---:|---:|---:|---:|---:|
|ARG  | 49.03| 48.62| 49.43| 36.99| 42.34| 55.5| 62.08| 9.44|
```

## Cálculo de medias y cuartiles por país

```

# figura 6
# unir datos y crear cluster únicos
# función
#
location_est <- function(data, var){
  # diseño (BRR)
  library(survey)
  erce_brr <- data %>%
    erce::remove_labels() %>%
    mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%
    mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, " ", STRATA)))) %>%
    mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, " ", ID SCHOOL)))) %>%
    mutate(id_l = seq(1:nrow(.)))
  #
  # función
  #
  ctry_acron <- data %>%
    dplyr::select(COUNTRY) %>%
    unique() %>%
    dplyr::pull()
  #
  # calcular tabla
  loc_est <- erce_brr %>%
    summarize(
      e = survey_mean({{var}}, na.rm=TRUE, vartype = 'ci'),
      q = survey_quantile({{var}}, na.rm=TRUE, c(0.10, 0.25, 0.75, .90)),
      sd = survey_sd({{var}}, na.rm = TRUE)
    ) %>%
    rename(l1 = e_low) %>%
    rename(u1 = e_up) %>%
    rename(p10 = q.q10) %>%
    rename(p25 = q.q25) %>%
    rename(p75 = q.q75) %>%
    rename(p90 = q.q90) %>%
    mutate(ctry = ctry_acron) %>%
    dplyr::select(ctry, e, ll, ul, p10, p25, p75, p90, sd
    )
  return(loc_est)
}

# definir dígitos en consola
#
options(digits = 10)
options(scipen = 999999)

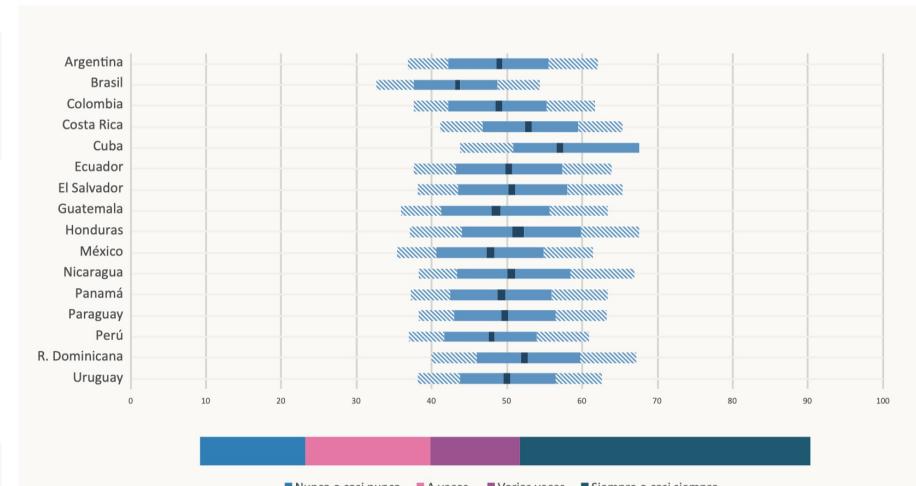
# resultado por país
#
dplyr::filter(erce_a6, COUNTRY == 'ARG') %>%
  location_est(data = ., var = AURES) %>%
  knitr::kable(., digits = 2)

|ctry |   e|    ll|    ul|   p10|   p25|   p75|   p90|    sd|
|----:|---:|---:|---:|---:|---:|---:|---:|---:|
|ARG | 49.03| 48.62| 49.43| 36.99| 42.34| 55.5| 62.08| 9.44|

```

- El cálculo de cuartiles, medias, intervalos de confianza, y desviación estándar de una variable continua, empleando diseño, implementar emplear diferentes pasos.

Figura 6. Distribución de puntajes en autorregulación escolar según país, junto al promedio regional



Nota: En las barras de los gráficos, el centro incluye un rectángulo oscuro que representa la media. Su ancho representa el intervalo de confianza de la media. Luego se muestra una barra que se extiende hacia la izquierda y la derecha que representa a los percentiles de 25 y 75, y, más allá, en los extremos más claros de esta misma barra se incluyen los percentiles 10 y 90.

# Cálculo de medias y cuartiles por país

```

# figura 6
# unir datos y crear cluster únicos
# función
#
location_est <- function(data, var){
  # diseño (BRR)
  library(srvyr)
  erce_brr <- data %>%
    erce::remove_labels() %>%
    mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%
    mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, " ", STRATA)))) %>%
    mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, " ", ID SCHOOL)))) %>%
    mutate(id_l = seq(1:nrow(.)))
  #
  # función
  #
  ctry_acron <- data %>%
    dplyr::select(COUNTRY) %>%
    unique() %>%
    dplyr::pull()
  #
  # calcular tabla
  loc_est <- erce_brr %>%
    summarize(
      e = survey_mean({{var}}, na.rm=TRUE, vartype = 'ci'),
      q = survey_quantile({{var}}, na.rm=TRUE, c(0.10, 0.25, 0.75, .90)),
      sd = survey_sd({{var}}, na.rm = TRUE)
    ) %>%
    rename(e_low = e_low) %>%
    rename(u_low = e_low) %>%
    rename(p10 = q_q10) %>%
    rename(p25 = q_q25) %>%
    rename(p75 = q_q75) %>%
    rename(p90 = q_q90) %>%
    mutate(ctry = ctry_acron) %>%
    dplyr::select(ctry, e, ll, ul, p10, p25, p75, p90, sd)
  )
  return(loc_est)
}

# definir dígitos en consola
#
options(digits = 10)
options(scipen = 999999)

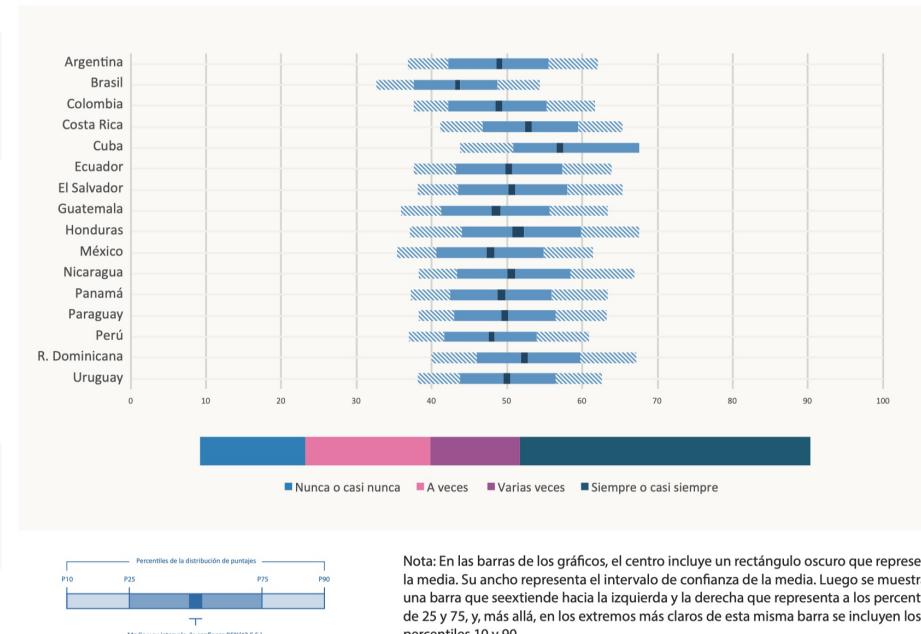
# resultado por país
#
dplyr::filter(erce_a6, COUNTRY == 'ARG') %>%
  location_est(data = ., var = AURES) %>%
  knitr::kable(., digits = 2)

|ctry |   e|   ll|   ul|   p10|   p25|   p75|   p90|   sd|
|----|---:|---:|---:|---:|---:|---:|---:|---:|
|ARG | 49.03| 48.62| 49.43| 36.99| 42.34| 55.5| 62.08| 9.44|

```

- El cálculo de cuartiles, medias, intervalos de confianza, y desviación estándar de una variable continua, empleando diseño, implementar emplear diferentes pasos.

Figura 6. Distribución de puntajes en autorregulación escolar según país, junto al promedio regional



- En este desarrollo, emplearemos un atajo. Este atajo consiste en crear una función, la cual nos ahorra líneas de código. Empleando este atajo, nos bastaría con crear una línea por país y podremos reproducir los resultados de la Figura 6.

## Cálculo de medias y cuartiles por país

```
# -----
# figura 6
# -----
# [... continua código anterior]

#-----
# crear y desplegar tabla
#-----

erce_a6 %>%
split(.$COUNTRY) %>%
purrr::map( ~ location_est(
data = .,
var = AURES
)) %>%
dplyr::bind_rows(., .id = 'COUNTRY') %>%
knitr::kable(., digits = 2)
```

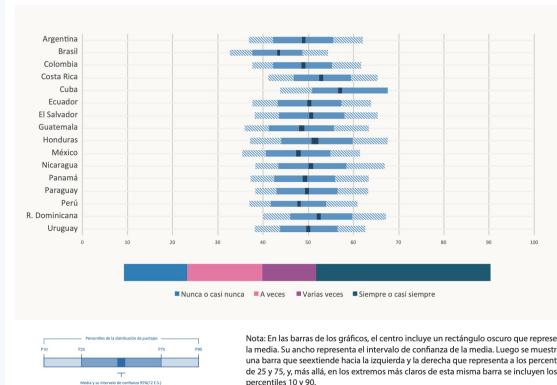
COUNTRY	ctry	e	ll	ul	p10	p25	p75	p90	sd
ARG	ARG	49.03	48.62	49.43	36.99	42.34	55.50	62.08	9.44
BRA	BRA	43.44	43.08	43.80	32.88	37.78	48.83	54.36	8.61
COL	COL	48.89	48.40	49.38	37.81	42.38	55.26	61.57	9.10
CRI	CRI	52.87	52.45	53.29	41.27	46.87	59.42	65.18	8.97
CUB	CUB	57.02	56.54	57.50	43.81	50.88	67.45	67.45	9.55
DOM	DOM	52.37	51.92	52.82	40.02	46.06	59.58	66.95	9.81
ECU	ECU	50.25	49.78	50.73	37.78	43.36	57.31	63.78	9.81
GTM	GTM	48.56	47.98	49.14	36.05	41.32	55.57	63.16	10.16
HND	HND	51.47	50.71	52.23	37.25	44.04	59.72	67.45	10.81
MEX	MEX	47.77	47.26	48.29	35.59	40.82	54.81	61.35	9.74
NIC	NIC	50.58	50.04	51.12	38.46	43.41	58.39	66.73	10.08
PAN	PAN	49.27	48.74	49.81	37.39	42.61	55.92	63.16	9.80
PER	PER	47.96	47.53	48.38	37.10	41.77	53.95	60.72	9.09
PRY	PRY	49.77	49.31	50.23	38.39	43.13	56.42	63.10	9.34
SLV	SLV	50.61	50.15	51.06	38.22	43.56	57.92	65.18	9.88
URY	URY	50.01	49.55	50.47	38.24	43.81	56.42	62.49	9.24

## Cálculo de medias y cuartiles por país

```
# -----
# figura 6
#
# [... continua código anterior]
#
# crear y desplegar tabla
#
erce_a6 %>%
split(.$COUNTRY) %>%
purrr::map(~ location_est(
data = .,
var = AURES
)) %>%
dplyr::bind_rows(., .id = 'COUNTRY') %>%
knitr::kable(., digits = 2)
```

COUNTRY	ctry	e	ll	ul	p10	p25	p75	p90	sd
ARG	ARG	49.03	48.62	49.43	36.99	42.34	55.50	62.08	9.44
BRA	BRA	43.44	43.08	43.80	32.88	37.78	48.83	54.36	8.61
COL	COL	48.89	48.40	49.38	37.81	42.38	55.26	61.57	9.10
CRI	CRI	52.87	52.45	53.29	41.27	46.87	59.42	65.18	8.97
CUB	CUB	57.02	56.54	57.50	43.81	50.88	67.45	67.45	9.55
DOM	DOM	52.37	51.92	52.82	40.02	46.06	59.58	66.95	9.81
ECU	ECU	50.25	49.78	50.73	37.78	43.36	57.31	63.78	9.81
GTM	GTM	48.56	47.98	49.14	36.05	41.32	55.57	63.16	10.16
HND	HND	51.47	50.71	52.23	37.25	44.04	59.72	67.45	10.81
MEX	MEX	47.77	47.26	48.29	35.59	40.82	54.81	61.35	9.74
NIC	NIC	50.58	50.04	51.12	38.46	43.41	58.39	66.73	10.08
PAN	PAN	49.27	48.74	49.81	37.39	42.61	55.92	63.16	9.80
PER	PER	47.96	47.53	48.38	37.10	41.77	53.95	60.72	9.09
PRY	PRY	49.77	49.31	50.23	38.39	43.13	56.42	63.10	9.34
SLV	SLV	50.61	50.15	51.06	38.22	43.56	57.92	65.18	9.88
URY	URY	50.01	49.55	50.47	38.24	43.81	56.42	62.49	9.24

Figura 6. Distribución de puntuajes en autorregulación escolar según país, junto al promedio regional



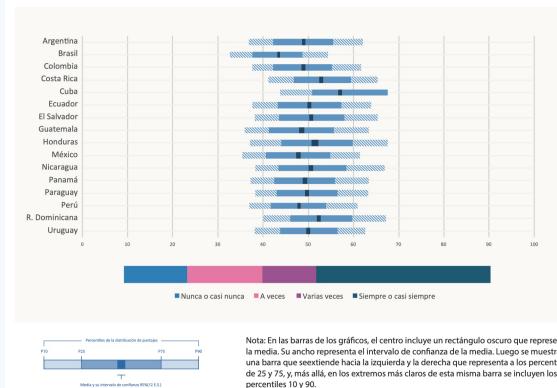
Nota: En las barras de los gráficos, el centro incluye un rectángulo oscuro que representa la media. Se muestra representando el intervalo de confianza de la media. Luego se muestra una barra que se extiende hacia la izquierda y la derecha que representa los percentiles de 25 y 75, y más allá, en los extremos más claros de esta misma barra se incluyen los percentiles 10 y 90.

## Cálculo de medias y cuartiles por país

```
# -----
# figura 6
#
# [... continua código anterior]
#
# crear y desplegar tabla
#
erce_a6 %>%
split(.$COUNTRY) %>%
purrr::map(~ location_est(
data = .,
var = AURES
)) %>%
dplyr::bind_rows(., .id = 'COUNTRY') %>%
knitr::kable(., digits = 2)
```

COUNTRY	ctry	e	ll	ul	p10	p25	p75	p90	sd
ARG	ARG	49.03	48.62	49.43	36.99	42.34	55.50	62.08	9.44
BRA	BRA	43.44	43.08	43.80	32.88	37.78	48.83	54.36	8.61
COL	COL	48.89	48.40	49.38	37.81	42.38	55.26	61.57	9.10
CRI	CRI	52.87	52.45	53.29	41.27	46.87	59.42	65.18	8.97
CUB	CUB	57.02	56.54	57.50	43.81	50.88	67.45	67.45	9.55
DOM	DOM	52.37	51.92	52.82	40.02	46.06	59.58	66.95	9.81
ECU	ECU	50.25	49.78	50.73	37.78	43.36	57.31	63.78	9.81
GTM	GTM	48.56	47.98	49.14	36.05	41.32	55.57	63.16	10.16
HND	HND	51.47	50.71	52.23	37.25	44.04	59.72	67.45	10.81
MEX	MEX	47.77	47.26	48.29	35.59	40.82	54.81	61.35	9.74
NIC	NIC	50.58	50.04	51.12	38.46	43.41	58.39	66.73	10.08
PAN	PAN	49.27	48.74	49.81	37.39	42.61	55.92	63.16	9.80
PER	PER	47.96	47.53	48.38	37.10	41.77	53.95	60.72	9.09
PRY	PRY	49.77	49.31	50.23	38.39	43.13	56.42	63.10	9.34
SLV	SLV	50.61	50.15	51.06	38.22	43.56	57.92	65.18	9.88
URY	URY	50.01	49.55	50.47	38.24	43.81	56.42	62.49	9.24

Figura 6. Distribución de puntuajes en autorregulación escolar según país, junto al promedio regional



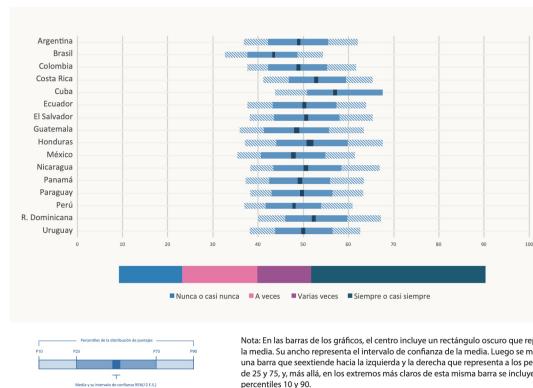
- Además de emplear un atajo mediante una función creada, empleamos dos trucos adicionales para ahorrarnos líneas de código. Los atajos empleados son las funciones de **split()** y **purrr::map()**.

## Cálculo de medias y cuartiles por país

```
# -----
# figura 6
#
# [... continua código anterior]
#
# crear y desplegar tabla
#
erce_a6 %>%
split(.$COUNTRY) %>%
purrr::map(~ location_est(
data = .,
var = AURES
)) %>%
dplyr::bind_rows(., .id = 'COUNTRY') %>%
knitr::kable(., digits = 2)
```

COUNTRY	ctry	e	ll	ul	p10	p25	p75	p90	sd
ARG	ARG	49.03	48.62	49.43	36.99	42.34	55.50	62.08	9.44
BRA	BRA	43.44	43.08	43.80	32.88	37.78	48.83	54.36	8.61
COL	COL	48.89	48.40	49.38	37.81	42.38	55.26	61.57	9.10
CRI	CRI	52.87	52.45	53.29	41.27	46.87	59.42	65.18	8.97
CUB	CUB	57.02	56.54	57.50	43.81	50.88	67.45	67.45	9.55
DOM	DOM	52.37	51.92	52.82	40.02	46.06	59.58	66.95	9.81
ECU	ECU	50.25	49.78	50.73	37.78	43.36	57.31	63.78	9.81
GTM	GTM	48.56	47.98	49.14	36.05	41.32	55.57	63.16	10.16
HND	HND	51.47	50.71	52.23	37.25	44.04	59.72	67.45	10.81
MEX	MEX	47.77	47.26	48.29	35.59	40.82	54.81	61.35	9.74
NIC	NIC	50.58	50.04	51.12	38.46	43.41	58.39	66.73	10.08
PAN	PAN	49.27	48.74	49.81	37.39	42.61	55.92	63.16	9.80
PER	PER	47.96	47.53	48.38	37.10	41.77	53.95	60.72	9.09
PRY	PRY	49.77	49.31	50.23	38.39	43.13	56.42	63.10	9.34
SLV	SLV	50.61	50.15	51.06	38.22	43.56	57.92	65.18	9.88
URY	URY	50.01	49.55	50.47	38.24	43.81	56.42	62.49	9.24

Figura 6. Distribución de puntuajes en autorregulación escolar según país, junto al promedio regional



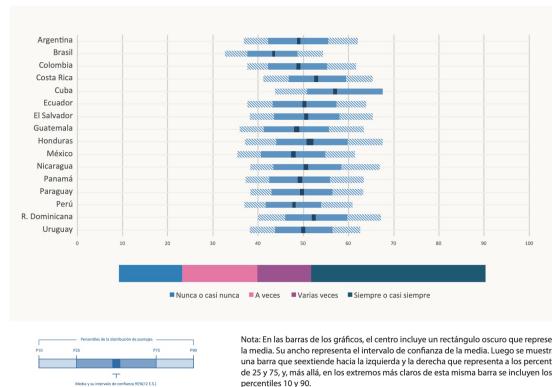
- Además de emplear un atajo mediante una función creada, empleamos dos trucos adicionales para ahorrarnos líneas de código. Los atajos empleados son las funciones de **split()** y **purrr::map()**.
- **split()** nos permite crear listas de bases de datos.

## Cálculo de medias y cuartiles por país

```
# -----
# figura 6
#
# [... continua código anterior]
#
# crear y desplegar tabla
#
erce_a6 %>%
split(.$COUNTRY) %>%
purrr::map(~ location_est(
data = .,
var = AURES
)) %>%
dplyr::bind_rows(., .id = 'COUNTRY') %>%
knitr::kable(., digits = 2)
```

COUNTRY	ctry	e	ll	ul	p10	p25	p75	p90	sd
ARG	ARG	49.03	48.62	49.43	36.99	42.34	55.50	62.08	9.44
BRA	BRA	43.44	43.08	43.80	32.88	37.78	48.83	54.36	8.61
COL	COL	48.89	48.40	49.38	37.81	42.38	55.26	61.57	9.10
CRI	CRI	52.87	52.45	53.29	41.27	46.87	59.42	65.18	8.97
CUB	CUB	57.02	56.54	57.50	43.81	50.88	67.45	67.45	9.55
DOM	DOM	52.37	51.92	52.82	40.02	46.06	59.58	66.95	9.81
ECU	ECU	50.25	49.78	50.73	37.78	43.36	57.31	63.78	9.81
GTM	GTM	48.56	47.98	49.14	36.05	41.32	55.57	63.16	10.16
HND	HND	51.47	50.71	52.23	37.25	44.04	59.72	67.45	10.81
MEX	MEX	47.77	47.26	48.29	35.59	40.82	54.81	61.35	9.74
NIC	NIC	50.58	50.04	51.12	38.46	43.41	58.39	66.73	10.08
PAN	PAN	49.27	48.74	49.81	37.39	42.61	55.92	63.16	9.80
PER	PER	47.96	47.53	48.38	37.10	41.77	53.95	60.72	9.09
PRY	PRY	49.77	49.31	50.23	38.39	43.13	56.42	63.10	9.34
SLV	SLV	50.61	50.15	51.06	38.22	43.56	57.92	65.18	9.88
URY	URY	50.01	49.55	50.47	38.24	43.81	56.42	62.49	9.24

Figura 6. Distribución de puntuajes en autorregulación escolar según país, junto al promedio regional



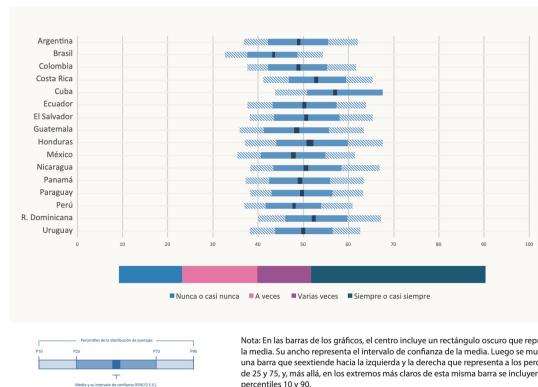
- Además de emplear un atajo mediante una función creada, empleamos dos trucos adicionales para ahorrarnos líneas de código. Los atajos empleados son las funciones de **split()** y **purrr::map()**.
- **split()** nos permite crear listas de bases de datos.
- **purrr::map()** nos permite aplicar funciones sobre una lista de bases de datos.

# Cálculo de medias y cuartiles por país

```
# -----
# figura 6
#
# [... continua código anterior]
#
# crear y desplegar tabla
#
erce_a6 %>%
split(.$COUNTRY) %>%
purrr::map(~ location_est(
data = .,
var = AURES
)) %>%
dplyr::bind_rows(., .id = 'COUNTRY') %>%
knitr::kable(., digits = 2)
```

COUNTRY	ctry	e	ll	ul	p10	p25	p75	p90	sd
ARG	ARG	49.03	48.62	49.43	36.99	42.34	55.50	62.08	9.44
BRA	BRA	43.44	43.08	43.80	32.88	37.78	48.83	54.36	8.61
COL	COL	48.89	48.40	49.38	37.81	42.38	55.26	61.57	9.10
CRI	CRI	52.87	52.45	53.29	41.27	46.87	59.42	65.18	8.97
CUB	CUB	57.02	56.54	57.50	43.81	50.88	67.45	67.45	9.55
DOM	DOM	52.37	51.92	52.82	40.02	46.06	59.58	66.95	9.81
ECU	ECU	50.25	49.78	50.73	37.78	43.36	57.31	63.78	9.81
GTM	GTM	48.56	47.98	49.14	36.05	41.32	55.57	63.16	10.16
HND	HND	51.47	50.71	52.23	37.25	44.04	59.72	67.45	10.81
MEX	MEX	47.77	47.26	48.29	35.59	40.82	54.81	61.35	9.74
NIC	NIC	50.58	50.04	51.12	38.46	43.41	58.39	66.73	10.08
PAN	PAN	49.27	48.74	49.81	37.39	42.61	55.92	63.16	9.80
PER	PER	47.96	47.53	48.38	37.10	41.77	53.95	60.72	9.09
PRY	PRY	49.77	49.31	50.23	38.39	43.13	56.42	63.10	9.34
SLV	SLV	50.61	50.15	51.06	38.22	43.56	57.92	65.18	9.88
URY	URY	50.01	49.55	50.47	38.24	43.81	56.42	62.49	9.24

Figura 6. Distribución de puntuajes en autorregulación escolar según país, junto al promedio regional



- Además de emplear un atajo mediante una función creada, empleamos dos trucos adicionales para ahorrarnos líneas de código. Los atajos empleados son las funciones de **split()** y **purrr::map()**.
- **split()** nos permite crear listas de bases de datos.
- **purrr::map()** nos permite aplicar funciones sobre una lista de bases de datos.
- Mezclando ambas funciones podemos ahorrarnos una serie de líneas de códigos, evitando escribir funciones país por país.

Poblacionales

# Regresiones

Reproducindo resultados de brechas (UNESCO-OREALC, 2021a, Figura 7)

## Modelos de regresiones por país

```
# -----  
# figura 7  
# -----  
  
# abrir datos y crear variables  
# -----  
  
erce_a6 <- erce_qa6 %>%  
  erce:::remove_labels() %>%  
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%  
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, " ", STRATA)))) %>%  
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, " ", IDSCHOOL)))) %>%  
  mutate(id_i = seq(1:nrow(.))) %>%  
  mutate(ses = ISECF) %>%  
  mutate(ses_g = r4sda::ic_wmean(x = ses, w = WS, j = id_k)) %>%  
  mutate(ses_m = ses - ses_g)  
  
#-----  
# crear base BRR para un país  
#-----  
  
library(srveyr)  
bra_brr <- erce_a6 %>%  
  dplyr::filter(COUNTRY == 'BRA') %>%  
  erce:::remove_labels() %>%  
  as_survey_reps(type = ' Fay',  
  repweights = starts_with(' BRR'),  
  weights = 'WT',  
  combined_weights = TRUE,  
  rho = .5,  
  mse = TRUE)  
  
#-----  
# ajustar modelo  
#-----  
  
reg_01 <- survey::svyglm(AURES ~ ses_m, design = bra_brr)  
  
#-----  
# definir dígitos en consola  
#-----  
  
options(digits = 10)  
options(scipen = 999999)  
  
#-----  
# mostrar tabla  
#-----  
  
broom::tidy(reg_01)%>%  
knitr::kable(., digits = 2)
```

term	estimate	std.error	statistic	p.value
(Intercept)	43.48	0.18	246.05	0
ses_m	2.17	0.19	11.52	0

# Modelos de regresiones por país

```

# -----#
# figura 7
# -----#
# abrir datos y crear variables
# -----#
erce_a6 <- erce_qa6 %>%
  erce:::remove_labels() %>%
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, " ", STRATA)))) %>%
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, " ", IDSCHOOL)))) %>%
  mutate(id_i = seq(1:nrow(.))) %>%
  mutate(ses = ISECF) %>%
  mutate(ses_g = r4sda::c_wmean(x = ses, w = WS, j = id_k)) %>%
  mutate(ses_m = ses - ses_g)

#-----#
# crear base BRR para un país
#-----#
library(srveyr)
bra_brr <- erce_a6 %>%
  dplyr::filter(COUNTRY == 'BRA') %>%
  erce:::remove_labels() %>%
  as_survey_rep(type = ' Fay',
  repweights = starts_with(' BRR'),
  weights = 'WT',
  combined_weights = TRUE,
  rho = .5,
  mse = TRUE)

#-----#
# ajustar modelo
#-----#
reg_01 <- survey::svyglm(AURES ~ ses_m, design = bra_brr)

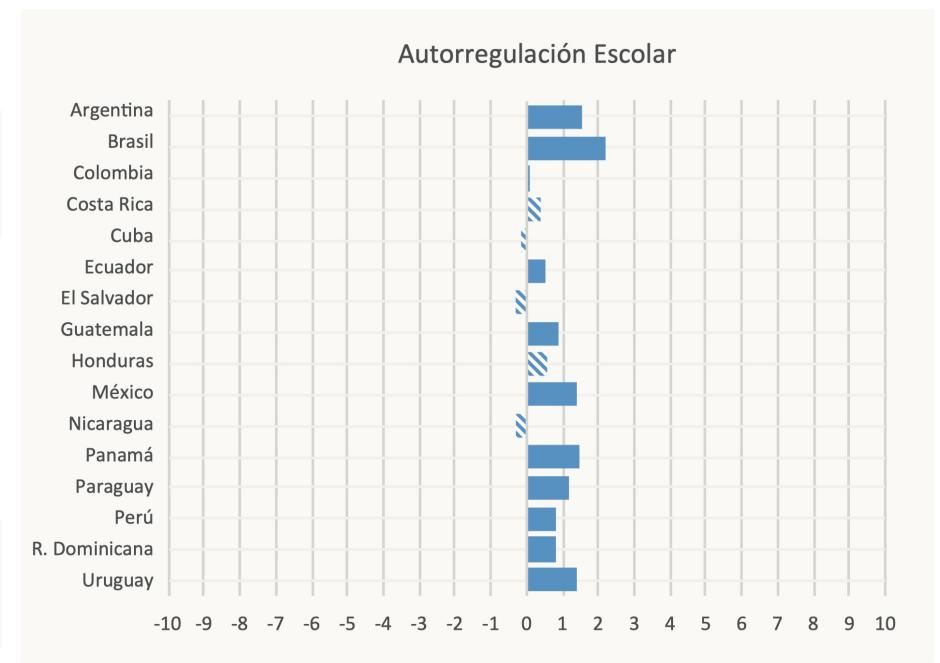
#-----#
# definir dígitos en consola
#-----#
options(digits = 10)
options(scipen = 999999)

#-----#
# mostrar tabla
#-----#
broom:::tidy(reg_01)%>%
knitr::kable(., digits = 2)

| term      | estimate | std.error | statistic | p.value |
|:-----:|:-----:|:-----:|:-----:|:-----:|
|(Intercept)| 43.48 | 0.18 | 246.05 | 0 |
|ses_m | 2.17 | 0.19 | 11.52 | 0 |

```

**Figura 7. Relación entre habilidades socioemocionales y el nivel socioeconómico de la familia de los estudiantes**



## Modelos de regresiones por país

```
# figura 7
# abrir datos y crear variables
#
erce_a6 <- erce_qa6 %>%
  erce:::remove_labels() %>%
  mutate(id_k = as.numeric(as.factor(paste0(IDCTRY)))) %>%
  mutate(id_s = as.numeric(as.factor(paste0(IDCTRY, " ", STRATA)))) %>%
  mutate(id_j = as.numeric(as.factor(paste0(IDCTRY, " ", IDSCHOOL)))) %>%
  mutate(id_i = seq(1:nrow(.))) %>%
  mutate(ses = ISECF) %>%
  mutate(ses_g = r4sda::c_wmean(x = ses, w = WS, j = id_k)) %>%
  mutate(ses_m = ses - ses_g)

# crear base BRR para un país
#
library(srvyr)
bra_brr <- erce_a6 %>%
  dplyr::filter(COUNTRY == 'BRA') %>%
  erce:::remove_labels() %>%
  as_survey_rep(type = ' Fay',
  repweights = starts_with(' BRR'),
  weights = 'WT',
  combined_weights = TRUE,
  rho = .5,
  mse = TRUE)

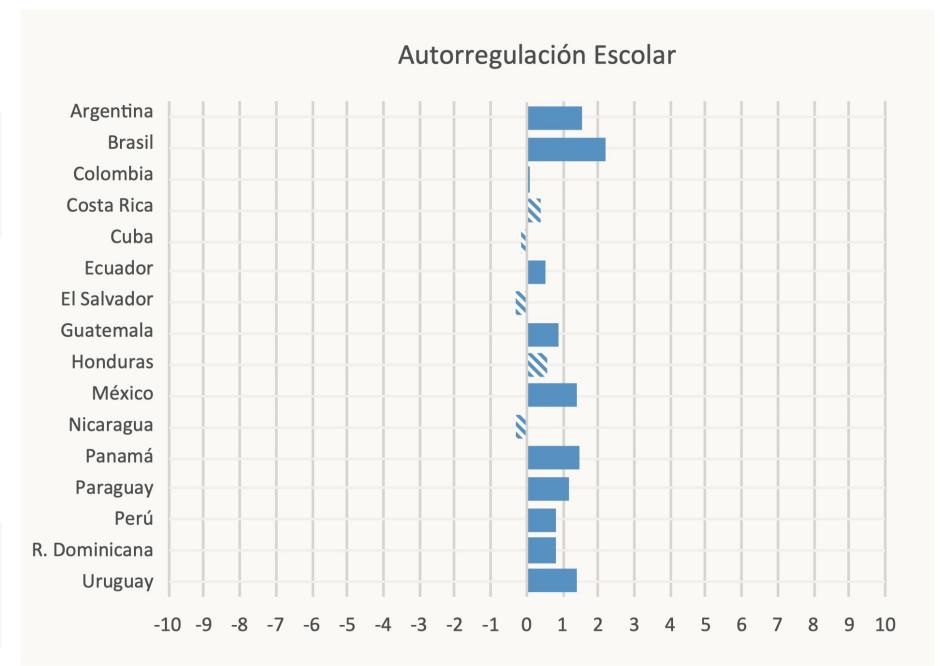
# ajustar modelo
#
reg_01 <- survey::svyglm(AURES ~ ses_m, design = bra_brr)

# definir dígitos en consola
#
options(digits = 10)
options(scipen = 999999)

# mostrar tabla
#
broom::tidy(reg_01)%>%
knitr::kable(., digits = 2)

|term      | estimate| std.error| statistic| p.value|
|:-----:|:-----:|:-----:|:-----:|:-----:|
|(Intercept)| 43.48| 0.18| 246.05| 0|
|ses_m| 2.17| 0.19| 11.52| 0|
```

Figura 7. Relación entre habilidades socioemocionales y el nivel socioeconómico de la familia de los estudiantes



- Reproducimos los resultados de **Brasil**. Para producir el resto de los resultados de los países, podemos emplear la misma lógica del ejemplo anterior: crear una función que genere los resultados sobre los que estamos interesados, y luego aplicamos **split** y **purrr::map**.

## Modelos de regresiones por país

```
# -----  
# figura 7  
# -----  
  
# abrir datos y crear variables  
# -----  
  
data_a6 <- erce_qa6 %>%  
  erce::remove_labels() %>%  
  mutate(id_k = as.numeric(as.factor(paste0(IDCNTRY)))) %>%  
  mutate(id_s = as.numeric(as.factor(paste0(IDCNTRY, " ", STRATA)))) %>%  
  mutate(id_j = as.numeric(as.factor(paste0(IDCNTRY, " ", IDSCHOOL)))) %>%  
  mutate(id_i = seq(1:nrow(.))) %>%  
  mutate(ses = ISECF) %>%  
  mutate(ses_g = r4sda::ic_wmean(x = ses, w = WS, j = id_k)) %>%  
  mutate(ses_m = ses - ses_g)  
  
# -----  
# función  
# -----  
  
reg_gap <- function(data, var){  
  library(srvyr)  
  erce_brr <- data %>%  
    mutate(y = {{var}}) %>%  
    erce::remove_labels() %>%  
    as_survey_rep(type = 'Fay',  
      repweights = starts_with('BRR'),  
      weights = 'WT',  
      combined_weights = TRUE,  
      rho = .5,  
      mse = TRUE)  
  
  ctry_acron <- data %>%  
    dplyr::select(COUNTRY) %>%  
    unique() %>%  
    dplyr::pull()  
  
  reg_est <- survey::svyglm(y ~ ses_m, design = erce_brr) %>%  
    broom::tidy() %>%  
    mutate(ctry = ctry_acron) %>%  
    dplyr::select(ctry, term, estimate, std.error, statistic, p.value)  
  
  return(reg_est)  
}  
}
```

# Modelos de regresiones por país

```

# figura 7
#
# abrir datos y crear variables
#
data_a6 <- erce_qa6 %>%
  erce::remove_labels() %>%
  mutate(id_k = as.numeric(as.factor(paste0(IDCTRY)))) %>%
  mutate(id_s = as.numeric(as.factor(paste0(IDCTRY, " ", STRATA)))) %>%
  mutate(id_j = as.numeric(as.factor(paste0(IDCTRY, " ", IDSCHOOL)))) %>%
  mutate(id_i = seq(1:nrow(.))) %>%
  mutate(ses = ISECF) %>%
  mutate(ses_g = r4sda::c_wmean(x = ses, w = WS, j = id_k)) %>%
  mutate(ses_m = ses - ses_g)

# función
#
reg_gap <- function(data, var){
  library(srveyr)
  erce_brr <- data %>%
    mutate(y = {{var}}) %>%
    erce::remove_labels() %>%
    as_survey_rep(type = 'Fay',
    repweights = starts_with('BRR'),
    weights = 'WT',
    combined_weights = TRUE,
    rho = .5,
    mse = TRUE)
  ctry_acron <- data %>%
    dplyr::select(COUNTRY) %>%
    unique() %>%
    dplyr::pull()
  reg_est <- survey::svyglm(y ~ ses_m, design = erce_brr) %>%
    broom::tidy() %>%
    mutate(ctry = ctry_acron) %>%
    dplyr::select(ctry, term, estimate, std.error, statistic, p.value)
  return(reg_est)
}

```

```

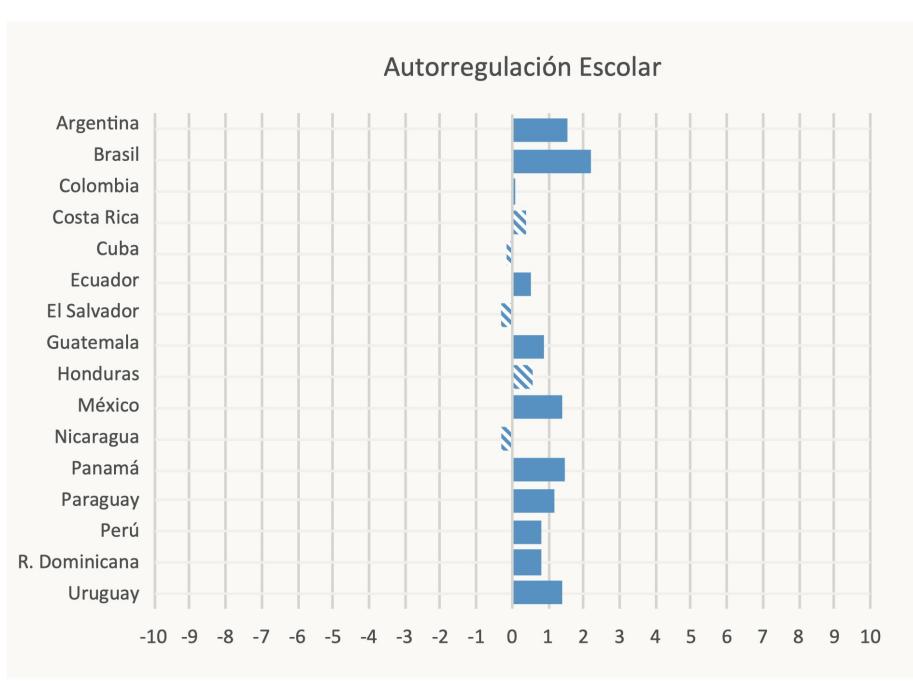
# figura 7
#
# [...] continua código anterior]
#
# definir dígitos en consola
#
options(digits = 10)
options(scipen = 999999)
#
# crear y desplegar tabla
#
data_a6 %>%
  split(.COUNTRY) %>%
  purrr::map(~ reg_gap(
    data = .,
    var = AURES
  )) %>%
  dplyr::bind_rows(., .id = 'COUNTRY') %>%
  dplyr::filter(term == 'ses_m') %>%
  knitr::kable(., digits = 2)

```

COUNTRY	ctry	term	estimate	std.error	statistic	p.value
ARG	ARG	ses_m	1.56	0.23	6.69	0.00
BRA	BRA	ses_m	2.17	0.19	11.52	0.00
COL	COL	ses_m	0.04	0.19	0.22	0.83
CRI	CRI	ses_m	0.39	0.21	1.80	0.08
CUB	CUB	ses_m	-0.16	0.25	-0.64	0.52
DOM	DOM	ses_m	0.81	0.24	3.45	0.00
ECU	ECU	ses_m	0.52	0.20	2.56	0.01
GTM	GTM	ses_m	0.87	0.21	4.20	0.00
HND	HND	ses_m	0.57	0.35	1.61	0.11
MEX	MEX	ses_m	1.38	0.28	5.01	0.00
NIC	NIC	ses_m	-0.31	0.31	-0.98	0.33
PAN	PAN	ses_m	1.44	0.19	7.75	0.00
PER	PER	ses_m	0.80	0.18	4.58	0.00
PRY	PRY	ses_m	1.17	0.24	4.83	0.00
SLV	SLV	ses_m	-0.31	0.18	-1.70	0.09
URY	URY	ses_m	1.41	0.15	9.18	0.00

## Modelos de regresiones por país

**Figura 7. Relación entre habilidades socioemocionales y el nivel socioeconómico de la familia de los estudiantes**



COUNTRY	ctry	term	estimate	std.error	statistic	p.value
ARG	ARG	ses_m	1.56	0.23	6.69	0.00
BRA	BRA	ses_m	2.17	0.19	11.52	0.00
COL	COL	ses_m	0.04	0.19	0.22	0.83
CRI	CRI	ses_m	0.39	0.21	1.80	0.08
CUB	CUB	ses_m	-0.16	0.25	-0.64	0.52
DOM	DOM	ses_m	0.81	0.24	3.45	0.00
ECU	ECU	ses_m	0.52	0.20	2.56	0.01
GTM	GTM	ses_m	0.87	0.21	4.20	0.00
HND	HND	ses_m	0.57	0.35	1.61	0.11
MEX	MEX	ses_m	1.38	0.28	5.01	0.00
NIC	NIC	ses_m	-0.31	0.31	-0.98	0.33
PAN	PAN	ses_m	1.44	0.19	7.75	0.00
PER	PER	ses_m	0.80	0.18	4.58	0.00
PRY	PRY	ses_m	1.17	0.24	4.83	0.00
SLV	SLV	ses_m	-0.31	0.18	-1.70	0.09
URY	URY	ses_m	1.41	0.15	9.18	0.00

- Con el conjunto de operaciones básicas revisadas, cualquiera de los resultados poblacionales presentes en los informes de ERCE 2019, se pueden reproducir.
- Lo importante en esta forma de trabajo, es identificar cuáles son los resultados esperados (qué estadístico se requiere producir, y en qué formato debe ser presentada la tabla), y luego generar cada uno de los pasos necesarios desde que se abre la base de datos, se preparan variables, se ajustan o calculan resultados, y se preparan las tablas esperadas.

# Muchas gracias!

## Referencias

Heeringa, S. G., West, B., & Berglund, P. A. (2009). Applied Survey Data Analysis. Taylor & Francis Group.

Rust, K. F., Krawchuk, S., & Monseur, C. (2017). Sample Design, Weighting, and Calculation of Sampling Variance. In P. Lietz, J. C. Cresswell, K. Rust, & R. J. Adams (Eds.), Implementation of Large-Scale Education Assessments (pp. 137–167). John Wiley & Sons, Ltd. [https://doi.org/10.1002/9781118762462\\_5](https://doi.org/10.1002/9781118762462_5)

UNESCO-OREALC. (2021a). Habilidades socioemocionales en América Latina y el Caribe: Estudio Regional Comparativo y Explicativo (ERCE 2019). <https://unesdoc.unesco.org/ark:/48223/pf0000380240>

Xie, Y. (2013). Dynamic Documents with R and knitr. In Dynamic Documents with R and knitr. Chapman and Hall/CRC. <http://www.crcpress.com/product/isbn/9781482203530>