

Análisis de datos como desarrollo

Descriptivos

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

Taller

Descriptivos

Generación de resultados descriptivos

Descriptivos

- Revisaremos diferentes ejemplos de descriptivos los cuales incluyen diferentes medidas de **locación**.
- Estos incluyen a:
 - Porcentajes
 - Medianas
 - Medias
- Revisaremos la generación de estos diferentes descriptivos, tanto para:
 - Escenarios de variables observadas
 - Escenarios que incluyen a variables con **valores plausibles**

Descriptivos

- Revisaremos diferentes ejemplos de descriptivos los cuales incluyen diferentes medidas de **locación**.
- Estos incluyen a:
 - Porcentajes
 - Medianas
 - Medias
- Revisaremos la generación de estos diferentes descriptivos, tanto para:
 - Escenarios de variables observadas
 - Escenarios que incluyen a variables con **valores plausibles**
- La generación de descriptivos, en el caso de los estudios de gran escala es de gran importancia.
- Estos resultados nos permiten comunicar, cual es el estado de cosas en una población, con un margen de error conocido.
- Empleando datos de ERCE 2019, es posible realizar inferencias a la población de estudiantes respecto a diferentes atributos, como por ejemplo:
 - repetición de grado
 - niveles de logro
 - logros en promedio

... entre varias otras preguntas. En las siguientes láminas, veremos una serie de ejemplos de generación de descriptivos empleando variables observadas, y luego, una serie de ejemplos, empleando variables representadas por valores plausibles.

Taller

Descriptivos

Descriptivos con variables observadas

Taller

Descriptivos

Generación de pesos replicados BRR con pesos senate

Código 1.1: pesos BRR con pesos estandarizados o pesos SENATE

- Las bases de datos actualmente solo incluyen los pesos replicados que emplean los pesos totales de cada país.
- Una forma de crear pesos replicados estandarizados, consiste en dividir el peso **BRR*****, por el peso total de la observación (**WT**).
- El resultado de esta operación nos brinda el factor de peso, de los pesos replicados sobre la observación.
- Podemos emplear este factor, y multiplicarlo por el peso *senate*, y así obtendremos pesos replicados **BRR**, pero en una escala diferente.
- Estamos realizando esta operación mediante la línea `mutate(repws001 = BRR1/WT * WS) %>%` para cada uno de los pesos replicados.
- Empleando estos pesos replicados, ahora podemos realizar análisis que emplean a los BRR, pero con pesos *senate* (`repws001`-`repws100`).
- Este paso previo es importante solo para la generación de resultados regionales. Esto nos permite producir cálculos de porcentajes, donde cada país pese de forma equivalente.

Código 1.1: pesos BRR con pesos estandarizados o pesos SENATE

- Las bases de datos actualmente solo incluyen los pesos replicados que emplean los pesos totales de cada país.
- Una forma de crear pesos replicados estandarizados, consiste en dividir el peso **BRR*****, por el peso total de la observación (**WT**).
- El resultado de esta operación nos brinda el factor de peso, de los pesos replicados sobre la observación.
- Podemos emplear este factor, y multiplicarlo por el peso **senate**, y así obtendremos pesos replicados **BRR**, pero en una escala diferente.
- Estamos realizando esta operación mediante la línea **mutate(repws001 = BRR1/WT * WS) %>%** para cada uno de los pesos replicados.
- Empleando estos pesos replicados, ahora podemos realizar análisis que emplean a los BRR, pero con pesos senate (**repws001-repws100**).
- Este paso previo es importante solo para la generación de resultados regionales. Esto nos permite producir cálculos de porcentajes, donde cada país pese de forma equivalente.

```
# -----  
# BRR con pesos senate  
# -----  
  
# -----  
# BRR/WT factor del BRR sobre los observados  
# -----  
  
data_a6 <- erce2019_qa6 %>%  
mutate(repws001 = BRR1/WT * WS) %>%  
mutate(repws002 = BRR2/WT * WS) %>%  
mutate(repws003 = BRR3/WT * WS) %>%  
mutate(repws004 = BRR4/WT * WS) %>%  
mutate(repws005 = BRR5/WT * WS) %>%  
mutate(repws006 = BRR6/WT * WS) %>%  
mutate(repws007 = BRR7/WT * WS) %>%  
mutate(repws008 = BRR8/WT * WS) %>%  
mutate(repws009 = BRR9/WT * WS) %>%  
mutate(repws010 = BRR10/WT * WS) %>%  
mutate(repws011 = BRR11/WT * WS) %>%  
mutate(repws012 = BRR12/WT * WS) %>%  
mutate(repws013 = BRR13/WT * WS) %>%  
mutate(repws014 = BRR14/WT * WS) %>%  
mutate(repws015 = BRR15/WT * WS) %>%  
mutate(repws016 = BRR16/WT * WS) %>%  
mutate(repws017 = BRR17/WT * WS) %>%  
mutate(repws018 = BRR18/WT * WS) %>%  
mutate(repws019 = BRR19/WT * WS) %>%  
mutate(repws020 = BRR20/WT * WS) %>%  
mutate(repws021 = BRR21/WT * WS) %>%  
mutate(repws022 = BRR22/WT * WS) %>%  
mutate(repws023 = BRR23/WT * WS) %>%  
mutate(repws024 = BRR24/WT * WS) %>%  
mutate(repws025 = BRR25/WT * WS) %>%  
mutate(repws026 = BRR26/WT * WS) %>%  
mutate(repws027 = BRR27/WT * WS) %>%  
mutate(repws028 = BRR28/WT * WS) %>%  
mutate(repws029 = BRR29/WT * WS) %>%  
mutate(repws030 = BRR30/WT * WS) %>%  
mutate(repws031 = BRR31/WT * WS) %>%  
mutate(repws032 = BRR32/WT * WS) %>%  
mutate(repws033 = BRR33/WT * WS) %>%  
mutate(repws034 = BRR34/WT * WS) %>%  
mutate(repws035 = BRR35/WT * WS) %>%  
mutate(repws036 = BRR36/WT * WS) %>%  
mutate(repws037 = BRR37/WT * WS) %>%  
mutate(repws038 = BRR38/WT * WS) %>%  
mutate(repws039 = BRR39/WT * WS) %>%  
mutate(repws040 = BRR40/WT * WS) %>%  
mutate(repws041 = BRR41/WT * WS) %>%  
mutate(repws042 = BRR42/WT * WS) %>%  
mutate(repws043 = BRR43/WT * WS) %>%  
mutate(repws044 = BRR44/WT * WS) %>%  
mutate(repws045 = BRR45/WT * WS) %>%  
mutate(repws046 = BRR46/WT * WS) %>%  
mutate(repws047 = BRR47/WT * WS) %>%  
mutate(repws048 = BRR48/WT * WS) %>%  
mutate(repws049 = BRR49/WT * WS) %>%  
mutate(repws050 = BRR50/WT * WS) %>%  
mutate(repws051 = BRR51/WT * WS) %>%  
mutate(repws052 = BRR52/WT * WS) %>%  
mutate(repws053 = BRR53/WT * WS) %>%  
mutate(repws054 = BRR54/WT * WS) %>%  
mutate(repws055 = BRR55/WT * WS) %>%  
mutate(repws056 = BRR56/WT * WS) %>%  
mutate(repws057 = BRR57/WT * WS) %>%  
mutate(repws058 = BRR58/WT * WS) %>%  
mutate(repws059 = BRR59/WT * WS) %>%  
mutate(repws060 = BRR60/WT * WS) %>%  
mutate(repws061 = BRR61/WT * WS) %>%  
mutate(repws062 = BRR62/WT * WS) %>%  
mutate(repws063 = BRR63/WT * WS) %>%  
mutate(repws064 = BRR64/WT * WS) %>%  
mutate(repws065 = BRR65/WT * WS) %>%  
mutate(repws066 = BRR66/WT * WS) %>%  
mutate(repws067 = BRR67/WT * WS) %>%  
mutate(repws068 = BRR68/WT * WS) %>%  
mutate(repws069 = BRR69/WT * WS) %>%  
mutate(repws070 = BRR70/WT * WS) %>%  
mutate(repws071 = BRR71/WT * WS) %>%  
mutate(repws072 = BRR72/WT * WS) %>%  
mutate(repws073 = BRR73/WT * WS) %>%  
mutate(repws074 = BRR74/WT * WS) %>%  
mutate(repws075 = BRR75/WT * WS) %>%  
mutate(repws076 = BRR76/WT * WS) %>%  
mutate(repws077 = BRR77/WT * WS) %>%  
mutate(repws078 = BRR78/WT * WS) %>%  
mutate(repws079 = BRR79/WT * WS) %>%  
mutate(repws080 = BRR80/WT * WS) %>%  
mutate(repws081 = BRR81/WT * WS) %>%  
mutate(repws082 = BRR82/WT * WS) %>%  
mutate(repws083 = BRR83/WT * WS) %>%  
mutate(repws084 = BRR84/WT * WS) %>%  
mutate(repws085 = BRR85/WT * WS) %>%  
mutate(repws086 = BRR86/WT * WS) %>%  
mutate(repws087 = BRR87/WT * WS) %>%  
mutate(repws088 = BRR88/WT * WS) %>%  
mutate(repws089 = BRR89/WT * WS) %>%  
mutate(repws090 = BRR90/WT * WS) %>%  
mutate(repws091 = BRR91/WT * WS) %>%  
mutate(repws092 = BRR92/WT * WS) %>%  
mutate(repws093 = BRR93/WT * WS) %>%  
mutate(repws094 = BRR94/WT * WS) %>%  
mutate(repws095 = BRR95/WT * WS) %>%  
mutate(repws096 = BRR96/WT * WS) %>%  
mutate(repws097 = BRR97/WT * WS) %>%  
mutate(repws098 = BRR98/WT * WS) %>%  
mutate(repws099 = BRR99/WT * WS) %>%  
mutate(repws100 = BRR100/WT * WS)
```

Taller

Descriptivos

Porcentajes regionales empleando BRR y pesos SENATE

Código 1.2: porcentaje con variable observada

- Vamos a reproducir los resultados de repetición de los estudiantes de la región.
 - Primero vamos a inspeccionar la variable original que produce los resultados
 - Luego, vamos a crear el objeto "survey", para que R puede estimar los errores estándar de los puntos estimados de interés.
 - Después, vamos a crear un objeto que contiene los resultados que necesitamos.
 - Finalmente, vamos a desplegar en pantalla los resultados obtenidos.

... en las siguientes láminas revisaremos estos pasos uno a uno.

Código 1.2: porcentaje con variable observada

- Vamos a reproducir los resultados de repetición de los estudiantes de la región.
 - Primero vamos a inspeccionar la variable original que produce los resultados
 - Luego, vamos a crear el objeto "survey", para que R puede estimar los errores estándar de los puntos estimados de interés.
 - Después, vamos a crear un objeto que contiene los resultados que necesitamos.
 - Finalmente, vamos a desplegar en pantalla los resultados obtenidos.

... en las siguientes láminas revisaremos estos pasos uno a uno.

```
# -----  
# estudiantes que han repetido  
# -----  
  
#-----  
# inspeccionar variable original  
#-----  
  
dplyr::count(data_a6, E6IT16, REPC)  
  
#-----  
# crear base BRR  
#-----  
  
library(srvyr)  
data_brr <- data_a6 %>%  
  erce::remove_labels() %>%  
  as_survey_rep(  
    type = 'Fay',  
    repweights = starts_with('repws'),  
    weights = 'WS',  
    combined_weights = TRUE,  
    rho = .5,  
    mse = TRUE  
  )  
  
# Opción: corrección a unidad primaria de muestreo que resulte  
# única al estrato  
  
library(survey)  
options(survey.lonely.psu="adjust")  
  
#-----  
# producir porcentaje regional  
#-----  
  
tabla_1 <- data_brr %>%  
  summarize(  
    proportion = survey_mean(REPC,na.rm=TRUE,  
    prop_method = 'logit',  
    vartype = "ci",  
    level = c(0.95)  
  )  
)  
  
#-----  
# mostrar tabla  
#-----  
  
knitr::kable(tabla_1, digits = 2)
```

Código 1.2: porcentaje con variable observada

- Vamos a reproducir los resultados de repetición de los estudiantes de las región.
 - **Primero vamos a inspeccionar la variable original que produce los resultados**
 - Luego, vamos a crear el objeto "survey", para que R puede estimar los errores estándar de los puntos estimados de interés.
 - Después, vamos a crear un objeto que contiene los resultados que necesitamos.
 - Finalmente, vamos a desplegar en pantalla los resultados obtenidos.

```
# -----  
# estudiantes que han repetido  
# -----  
  
#-----  
# inspeccionar variable original  
#-----  
  
dplyr::count(data_a6, E6IT16, REPC)  
  
> dplyr::count(data_a6, E6IT16, REPC)  
# A tibble: 5 × 3  
   E6IT16      <dbl+lbl>    REPC      <dbl+lbl>     n  
1 [Nunca he repetido.] 0 [Nunca] 63958  
2 [Una vez.] 1 [Una o más veces] 9403  
3 [Dos veces o más.] 1 [Una o más veces] 2504  
4 [No sé, no recuerdo.] NA NA 1001  
5 NA NA 3961
```

Código 1.2: porcentaje con variable observada

- Vamos a reproducir los resultados de repetición de los estudiantes de las región.
 - Primero vamos a inspeccionar la variable original que produce los resultados
 - **Luego, vamos a crear el objeto "survey", para que R puede estimar los errores estándar de los puntos estimados de interés.**
 - Después, vamos a crear un objeto que contiene los resultados que necesitamos.
 - Finalmente, vamos a desplegar en pantalla los resultados obtenidos.

```
# -----  
# estudiantes que han repetido  
# -----  
  
#-----  
# inspeccionar variable original  
#-----  
  
dplyr::count(data_a6, E6IT16, REPC)  
  
#-----  
# crear base BRR  
#-----  
  
library(srvyr)  
data_brr <- data_a6 %>%  
  erce::remove_labels() %>%  
  as_survey_rep(  
    type = 'Fay',  
    repweights = starts_with('repws'),  
    weights = 'WS',  
    combined_weights = TRUE,  
    rho = .5,  
    mse = TRUE  
  )  
  # Opción: corrección a unidad primaria de muestreo que resulte  
  # única al estrato  
  
library(survey)  
options(survey.lonely.psu="adjust")  
  
# Este código no nos genera un output.
```

Código 1.2: porcentaje con variable observada

- Vamos a reproducir los resultados de repetición de los estudiantes de las región.
 - Primero vamos a inspeccionar la variable original que produce los resultados
 - Luego, vamos a crear el objeto "survey", para que R puede estimar los errores estándar de los puntos estimados de interés.
 - **Después, vamos a crear un objeto que contiene los resultados que necesitamos.**
 - Finalmente, vamos a desplegar en pantalla los resultados obtenidos.

```
# -----  
# estudiantes que han repetido  
# -----  
  
#-----  
# inspeccionar variable original  
#-----  
  
dplyr::count(data_a6, E6IT16, REPC)  
  
#-----  
# crear base BRR  
#-----  
  
library(srvyr)  
data_brr <- data_a6 %>%  
  erce::remove_labels() %>%  
  as_survey_rep(  
    type = 'Fay',  
    repweights = starts_with('repws'),  
    weights = 'WS',  
    combined_weights = TRUE,  
    rho = .5,  
    mse = TRUE  
  )  
# Opción: corrección a unidad primaria de muestreo que resulte  
# única al estrato  
  
library(survey)  
options(survey.lonely.psu="adjust")  
  
#-----  
# producir porcentaje regional  
#-----  
  
tabla_1 <- data_brr %>%  
  summarise(  
    proportion = survey_mean(REPC,na.rm=TRUE,  
    prop_method = 'logit',  
    vartype = "ci",  
    level = c(0.95)  
  )  
)
```

Código 1.2: porcentaje con variable observada

- Vamos a reproducir los resultados de repetición de los estudiantes de la región.
 - Primero vamos a inspeccionar la variable original que produce los resultados
 - Luego, vamos a crear el objeto "survey", para que R puede estimar los errores estándar de los puntos estimados de interés.
 - Después, vamos a crear un objeto que contiene los resultados que necesitamos.
 - **Finalmente, vamos a desplegar en pantalla los resultados obtenidos.**

Output 1.2: proporción de estudiantes que han repetido algún grado (Estudiantes de Sexto Grado, ERCE 2019)

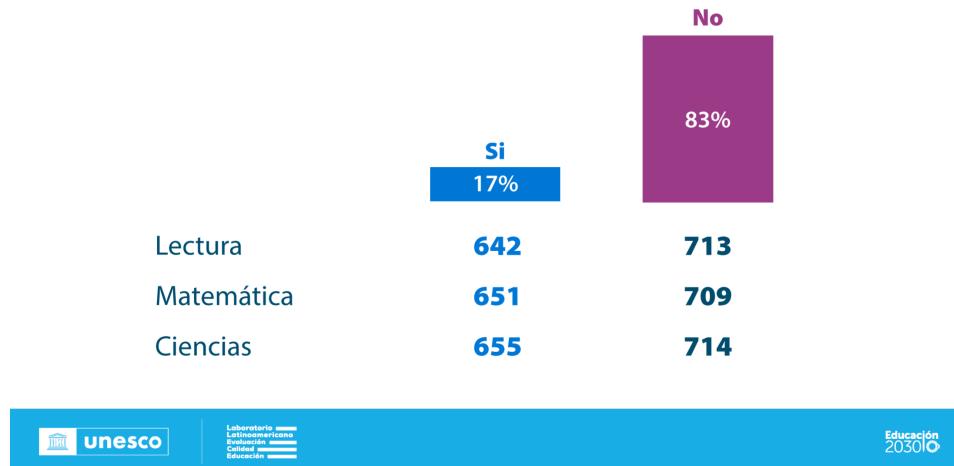
```
>knitr::kable(tabla_1, digits = 2)
```

proportion	proportion_low	proportion_upp
0.17	0.17	0.18

```
# -----  
# estudiantes que han repetido  
# -----  
  
#-----  
# inspeccionar variable original  
#-----  
  
dplyr::count(data_a6, E6IT16, REPC)  
  
#-----  
# crear base BRR  
#-----  
  
library(srvyr)  
data_brr <- data_a6 %>%  
  erce::remove_labels() %>%  
  as_survey_rep(  
    type = 'Fay',  
    repweights = starts_with('repws'),  
    weights = 'WS',  
    combined_weights = TRUE,  
    rho = .5,  
    mse = TRUE  
  )  
  
# Opción: corrección a unidad primaria de muestreo que resulte  
# única al estrato  
  
library(survey)  
options(survey.lonely.psu="adjust")  
  
#-----  
# producir porcentaje regional  
#-----  
  
tabla_1 <- data_brr %>%  
  summarize(  
    proportion = survey_mean(REPC,na.rm=TRUE,  
    prop_method = 'logit',  
    vartype = "ci",  
    level = c(0.95)  
  )  
  
#-----  
# mostrar tabla  
#-----  
  
knitr::kable(tabla_1, digits = 2)
```

Código 1.2: porcentaje con variable observada

Repetición de grado (6º)



La rutina del código 1.2 reproduce los resultados presentados en UNESCO (2021)
"Presentación de Resultados de factores Asociados".

```
# -----  
# estudiantes que han repetido  
# -----  
  
#-----  
# inspeccionar variable original  
#-----  
  
dplyr::count(data_a6, E6IT16, REPC)  
  
#-----  
# crear base BRR  
#-----  
  
library(srvyr)  
data_brr <- data_a6 %>%  
  erce::remove_labels() %>%  
  as_survey_rep()  
  type = 'Fay',  
  repweights = starts_with('repws'),  
  weights = 'WS',  
  combined_weights = TRUE,  
  rho = .5,  
  mse = TRUE  
)  
  
# Opción: corrección a unidad primaria de muestreo que resulte  
# única al estrato  
  
library(survey)  
options(survey.lonely.psu="adjust")  
  
#-----  
# producir porcentaje regional  
#-----  
  
tabla_1 <- data_brr %>%  
  summarize(  
    proportion = survey_mean(REPC,na.rm=TRUE,  
    prop_method = 'logit',  
    vartype = "ci",  
    level = c(0.95)  
  )  
)  
  
#-----  
# mostrar tabla  
#-----  
  
knitr::kable(tabla_1, digits = 2)
```

proportion	proportion_low	proportion_upp
0.17	0.17	0.18

Código 1.2: porcentaje con variable observada

- Vamos a reproducir los resultados de repetición de los estudiantes de las región.
 - Primero vamos a inspeccionar la variable original que produce los resultados
 - Luego, vamos a crear el objeto "survey", para que R puede estimar los errores estándar de los puntos estimados de interés.
 - Después, vamos a crear un objeto que contiene los resultados que necesitamos.
 - Finalmente, vamos a desplegar en pantalla los resultados obtenidos.

Nota: este tipo de rutina, con mínimas variantes, la aplicaremos sobre todos los tipos de descriptivos. En general, vamos a cargar **datos, preparar** los datos, crear objetos con **diseño, calcularemos** resultados, y luego desplegaremos los **resultados** obtenidos.

```
# -----  
# estudiantes que han repetido  
# -----  
  
#-----  
# inspeccionar variable original  
#-----  
  
dplyr::count(data_a6, E6IT16, REPC)  
  
#-----  
# crear base BRR  
#-----  
  
library(srvyr)  
data_brr <- data_a6 %>%  
  erce::remove_labels() %>%  
  as_survey_rep(  
    type = 'Fay',  
    repweights = starts_with('repws'),  
    weights = 'WS',  
    combined_weights = TRUE,  
    rho = .5,  
    mse = TRUE  
  )  
  
# Opción: corrección a unidad primaria de muestreo que resulte  
# única al estrato  
  
library(survey)  
options(survey.lonely.psu="adjust")  
  
#-----  
# producir porcentaje regional  
#-----  
  
tabla_1 <- data_brr %>%  
  summarize(  
    proportion = survey_mean(REPC,na.rm=TRUE,  
    prop_method = 'logit',  
    vartype = "ci",  
    level = c(0.95)  
  )  
  )  
  
#-----  
# mostrar tabla  
#-----  
  
knitr::kable(tabla_1, digits = 2)
```

Taller

Descriptivos

Porcentajes regionales empleando TSL y pesos SENATE

Código 1.3: porcentaje con variable observada (TSL)

- Vamos a reproducir los resultados de repetición de los estudiantes de la región.
 - Creamos variables de clustering comunes (id_k, id_s, id_j, id_i).
 - Creamos el objeto "survey", para que R puede estimar los errores estándar de los puntos estimados de interés, pero en esta ocasión, emplearemos Taylor Series linearization (TSL) en la librería **svy**
 - Calculamos los resultados
 - Finalmente desplegamos los resultados obtenidos.

Output 1.3 proporción de estudiantes que han repetido algún grado (Estudiantes de Sexto Grado, ERCE 2019)

```
>knitr::kable(tabla_2, digits = 2)
| proportion| proportion_low| proportion_upp|
|-----:|-----:|-----:|
| 0.17| 0.17| 0.18|
```

```
# -----
# estudiantes que han repetido
# ----

#-----
# unir datos y crear cluster únicos
#-----

erce_a6 <- erce:::erce_2019_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(.)))

#-----
# base de datos con diseño
#-----

# survey method: taylor series linearization library(svy)
library(svy)
data_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")

#-
# producir porcentaje regional
#-

tabla_2 <- data_tsl %>%
summarize(
proportion = survey_mean(REPC,na.rm=TRUE,
prop_method = 'logit',
vartype = "ci",
level = c(0.95)
)
)

#-
# mostrar tabla
#-

knitr::kable(tabla_2, digits = 2)
```

Taller

Porcentajes

Comparación de estimados con TSL y BRR

Notas: sobre la estimación de porcentajes, y las correcciones BRR y TSL

- Con pocos dígitos en los puntos estimados, en la mayoría de las ocasiones, los resultados generados con **BRR** o con **TSL**, van a ser muy similares.
 - Históricamente **BRR** es favorecido en los estudios de gran escala (Rust et al., 2017).
 - Sin embargo, las **diferencias sustantivas** que generan ambos métodos son **muy pequeñas** (ver Heeringa et al., 2009).
 - No obstante, se espera que la estimación de errores sobre cuartiles, incluyendo a la mediana (e.g., P25, P50, P75), sean más precisos con **BRR** (Rust et al., 2017, p164).
- Para este ejemplo, las diferencias se observan a partir del quinto decimal de los intervalos de confianza. En contraste, los puntos estimados son equivalentes.
 - La estimación de errores con BRR o TSL solo afecta a las medidas de incertidumbre alrededor de los puntos estimados (errores, e intervalos de confianza); y no a los puntos estimados.
- Para el caso de la estimación de porcentajes, es muy importante es el **método de estimación de errores**. Los porcentajes no poseen errores estándar simétricos y, mientras más se acerca un porcentaje al cero, su límite inferior esperado no puede ser menor a cero. En este caso, estamos empleando el método *logit* para obtener errores estándar asimétricos.

```
# -----  
# estudiantes que han repetido  
# -----  
  
#-----  
# mostrar tabla con dos dígitos  
#-----  
  
knitr::kable(tabla_1, digits = 2) # BRR  
knitr::kable(tabla_2, digits = 2) # TSL  
  
# BRR  
> knitr::kable(tabla_1, digits = 2)  
| proportion| proportion_low| proportion_upp|  
|-----:|-----:|-----:  
| 0.17| 0.17| 0.18|  
  
# TSL  
> knitr::kable(tabla_2, digits = 2)  
| proportion| proportion_low| proportion_upp|  
|-----:|-----:|-----:  
| 0.17| 0.17| 0.18|  
  
# -----  
# estudiantes que han repetido  
# -----  
  
#-----  
# mostrar tabla con cinco dígitos  
#-----  
  
knitr::kable(tabla_1, digits = 5) # BRR  
knitr::kable(tabla_2, digits = 5) # TSL  
  
# BRR  
> knitr::kable(tabla_1, digits = 5)  
| proportion| proportion_low| proportion_upp|  
|-----:|-----:|-----:  
| 0.17044| 0.16556| 0.17532|  
  
# TSL  
> knitr::kable(tabla_2, digits = 5)  
| proportion| proportion_low| proportion_upp|  
|-----:|-----:|-----:  
| 0.17044| 0.16549| 0.17539|
```

Taller

Porcentajes

Generación de porcentajes para un sólo país

Código 1.4: porcentaje con variable observada, para un solo país (BRR)

- Para producir los resultados de un solo país, basta con repetir los pasos anteriores, agregando una línea adicional.
- Esta línea adicional consiste en filtrar los casos del país de interés, y luego crear el objeto con el diseño muestral.

```
dplyr::filter(COUNTRY == 'URY') %>%
```

- De esta forma los pasos empleados son:
 - cargar los datos
 - filtrar por país de interés
 - crear el objeto de datos con diseño muestral
 - incluir la corrección para cluster solitarios en estratos (si es que los hubiera)
 - crear la tabla de porcentajes empleando la variable de interés
 - desplegar la tabla

```
# -----  
# estudiantes que han repetido  
# -----  
  
#-----  
# crear base BRR para un país  
#-----  
  
library(srvyr)  
ury_brr <- data_a6 %>%  
  erce::remove_labels() %>%  
  dplyr::filter(COUNTRY == 'URY') %>%  
  as_survey_rep()  
  type = 'Fay',  
  repweights = starts_with('repws'),  
  weights = 'WS',  
  combined_weights = TRUE,  
  rho = .5,  
  mse = TRUE  
 )  
  
# Opción: corrección a unidad primaria de muestreo que resulte  
# única al estrato  
  
library(survey)  
options(survey.lonely.psu="adjust")  
  
#-----  
# producir porcentaje regional  
#-----  
  
tabla_3 <- ury_brr %>%  
  summarize(  
    proportion = survey_mean(REPC, na.rm=TRUE),  
    prop_method = 'logit',  
    vartype = "ci",  
    level = c(0.95)  
  )  
)  
  
#-----  
# mostrar tabla  
#-----  
  
knitr::kable(tabla_3, digits = 2)
```

proportion	proportion_low	proportion_upp
0.19	0.17	0.21

Código 1.4: porcentaje con variable observada, para un solo país (BRR)

Uruguay: Reporte nacional de resultados

3.5. Descriptivos

A continuación, se incluyen los resultados descriptivos de los factores asociados incluidos en este reporte. Estos resultados se producen como estimados a la población, incluyendo medias y porcentajes según el tipo de factor asociado respectivo.

Tabla 12. Descriptivas poblaciones de los puntajes de factores asociados al logro de las pruebas del ERCE 2019 (medias y porcentajes).

Factores asociados a los estudiantes y sus familias	3º grado			6º grado		
	E	LI	LS	E	LI	LS
Nivel socioeconómico de la familia	0.67	0.62	0.73	0.66	0.61	0.71
dRepitencia	0.15	0.14	0.17	0.19	0.17	0.21
dInasistencia a la escuela	0.43	0.40	0.46	0.48	0.46	0.51
dDías de estudio a la semana	0.75	0.73	0.77	0.68	0.66	0.70
Involucramiento parental en el aprendizaje	-0.12	-0.16	-0.09	-0.06	-0.10	-0.02
dExpectativas educativas de los padres	0.58	0.55	0.60	0.56	0.53	0.59
Expectativas educativas de los profesores	0.19	0.15	0.25	0.18	0.14	0.23
Interés de los docentes por el bienestar de los estudiantes	0.24	0.18	0.29	0.14	0.07	0.20
Apoyo al aprendizaje de los estudiantes						
Lenguaje y Matemática	0.12	0.08	0.17			
Lenguaje				0.04	-0.02	0.09
Matemática				0.11	0.06	0.16
Ciencias				0.06	0.00	0.11
Organización de la enseñanza						
Lenguaje y Matemática	-0.05	-0.09	-0.01			
Lenguaje				-0.23	-0.29	-0.17
Matemática				-0.21	-0.27	-0.16
Ciencias				-0.21	-0.26	-0.16
Proceso escolar y prácticas docentes						
Lenguaje y Matemática	0.30	0.25	0.36			
Lenguaje				0.29	0.23	0.36
Matemática				0.33	0.24	0.41
Ciencias				0.29	0.22	0.36
Factor de escuela						
Nivel socioeconómico de la escuela	0.67	0.61	0.72	0.65	0.60	0.71
Escuela en lugar urbano (10 mil o más habitantes)	0.52	0.45	0.58	0.52	0.46	0.57

Nota: * Variables dicotómicas, para las cuales se reporta el porcentaje esperado a la población. El resto de los estimados corresponden a medias estimadas a la población. = estimado, LI = límite inferior del intervalo de confianza de 95 % del estimado reportado; LS = límite superior del intervalo de confianza de 95 % del estimado reportado.

```
# -----#
# estudiantes que han repetido
# -----#
#-----#
# crear base BRR para un país
#-----#
library(srvyr)
ury_brr <- data_a6 %>%
  erce:::remove_labels() %>%
  dplyr:::filter(COUNTRY == 'URY') %>%
  as_survey_rep()
type = 'Fay',
repweights = starts_with('repws'),
weights = 'WS',
combined_weights = TRUE,
rho = .5,
mse = TRUE
)
# Opción: corrección a unidad primaria de muestreo que resulte
# única al estrato
library(survey)
options(survey.lonely.psu="adjust")
#-----#
# producir porcentaje regional
#-----#
tabla_3 <- ury_brr %>%
  summarize(
    proportion = survey_mean(REPC, na.rm=TRUE),
    prop_method = 'logit',
    vartype = "ci",
    level = c(0.95)
  )
knitr::kable(tabla_3, digits = 2)

| proportion | proportion_low | proportion_upp |
|-----:|-----:|-----:|
| 0.19 | 0.17 | 0.21 |
```

La rutina del código 1.4 reproduce los resultados presentes en UNESCO (2021) "Estudio Regional Comparativo y Explicativo (ERCE 2019). Reporte nacional de resultados. Uruguay".

Taller

Medianas

Estimación de medianas y otros percentiles

Código 1.5: percentiles para un país (BRR)

- Para este ejemplo, empleamos el indicador de nivel socioeconómico de las familias de los estudiantes (**ISECF**).
- Al igual que en el código anterior, filtramos los casos del país de interés.

```
dplyr::filter(COUNTRY == 'URY') %>%
```

- De esta forma los pasos empleados son:

- cargar los datos
- filtrar por país de interés
- crear el objeto de datos con diseño muestral
- incluir la corrección para cluster solitarios en estratos (si es que los hubiera)
- aplicar las funciones que producen percentiles
- desplegar la tabla

```
# medianas y percentiles de un país
# ---

# crear base BRR para un país
# ---

library(srvyr)
ury_brr <- data_a6 %>%
  erce::remove_labels() %>%
  dplyr::filter(COUNTRY == 'URY') %>%
  as_survey_rep(
    type = 'Fay',
    repweights = starts_with('repws'),
    weights = 'WS',
    combined_weights = TRUE,
    rho = .5,
    mse = TRUE
  )

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

#--#
# producir percentiles
#--#

tabla_4 <- ury_brr %>%
  summarize(
    per = survey_quantile(ISECF,na.rm=TRUE,
      quantiles = c(.25, .50, .75),
      vartype = "ci",
      level = c(0.95)
    )
  )

#--#
# mostrar tabla
#--#

tabla_4 %>%
  knitr::kable(., digits = 2)
```

per_q25	per_q50	per_q75	per_q25_low	per_q50_low	per_q75_low	per_q25_upp	per_q50_upp	per_q75_upp
0.05	0.63	1.24	-0.01	0.58	1.18	0.12	0.7	1.28

Código 1.6: medianas y percentil 50 para un país (BRR)

- Para este ejemplo, empleamos el indicador de nivel socioeconómico de las familias de los estudiantes (**ISECF**).
- Al igual que en el código anterior, filtramos los casos del país de interés.

```
dplyr::filter(COUNTRY == 'URY') %>%
```

- De esta forma los pasos empleados son:

- cargar los datos
- filtrar por país de interés
- crear el objeto de datos con diseño muestral
- incluir la corrección para cluster solitarios en estratos (si es que los hubiera)
- aplicar las funciones que producen percentiles y medianas
- editamos la tabla para que sea más legible
- desplegar la tabla

```
# -  
# medianas y percentiles de un país  
# ---  
  
#-----  
# crear base BRR para un país  
#-----  
  
library(srvyr)  
ury_brr <- data_a6 %>%  
  erce::remove_labels() %>%  
  dplyr::filter(COUNTRY == 'URY') %>%  
  as_survey_rep(  
    type = 'Fay',  
    repweights = starts_with('repws'),  
    weights = 'WS',  
    combined_weights = TRUE,  
    rho = .5,  
    mse = TRUE  
  )  
  
# Opción: corrección a unidad primaria de muestreo que resulte  
# única al estrato  
  
library(survey)  
options(survey.lonely.psu="adjust")  
  
#-----  
# producir percentil 50 y mediana  
#-----  
  
tabla_5 <- ury_brr %>%  
  summarize(  
    per = survey_quantile(ISECF,na.rm=TRUE,  
      quantiles = c(.50),  
      vartype = "ci",  
      level = c(0.95)  
    ),  
    med = survey_median(ISECF,na.rm=TRUE,  
      vartype = "ci",  
      level = c(0.95)  
    )  
  )  
  
#-----  
# mostrar tabla  
#-----  
  
tabla_5 %>%  
  tidyverse::pivot_longer(  
    names(),  
    names_to = 'quantiles',  
    values_to = 'values'  
  ) %>%  
  arrange(quantiles) %>%  
  knitr::kable(., digits = 2)  
  
# Nota: estamos "volteando" la tabla con  
#       tidyverse::pivot_longer,  
#       para que sea más legible.
```

quantiles	values
med	0.63
med_low	0.58
med_upp	0.70
per_q50	0.63
per_q50_low	0.58
per_q50_upp	0.70

Taller

Medias

Estimación de medias de variables observadas

Código 1.7: medias de varios países (BRR)

- Para este ejemplo, empleamos el indicador de nivel socioeconómico de las familias de los estudiantes (**ISECF**).
- De esta forma los pasos empleados son:
 - cargar los datos
 - crear el objeto de datos con diseño muestral
 - incluir la corrección para cluster solitarios en estratos (si es que los hubiera)
 - aplicar las funciones que calculan medias empleando diseño
 - editamos la tabla para que sea más legible
 - desplegar la tabla

```
# -  
# medianas y percentiles de un país  
# ---  
  
#-----  
# crear base BRR para la region  
#-----  
  
library(srvyr)  
erce_brr <- data_a6 %>%  
  erce::remove_labels() %>%  
  as_survey_rep()  
  type = 'Fay',  
  repweights = starts_with('repws'),  
  weights = 'WS',  
  combined_weights = TRUE,  
  rho = .5,  
  mse = TRUE  
 )  
  
# Opción: corección a unidad primaria de muestreo que resulte  
# única al estrato  
  
library(survey)  
options(survey.lonely.psu="adjust")  
  
#-----  
# producir medias para cada país  
#-----  
  
tabla_6 <- erce_brr %>%  
  group_by(COUNTRY) %>%  
  summarize(  
    mean = survey_mean(ISECF,na.rm=TRUE,  
    vartype = "ci",  
    level = c(0.95)  
  )  
)  
  
#-----  
# mostrar tabla  
#-----  
  
tabla_6 %>%  
arrange(mean) %>%  
knitr::kable(., digits = 2)
```

COUNTRY	mean	mean_low	mean_upp
NIC	-0.64	-0.69	-0.60
HND	-0.52	-0.61	-0.44
GTM	-0.47	-0.54	-0.40
SLV	-0.28	-0.34	-0.22
PER	-0.21	-0.28	-0.14
PAN	-0.12	-0.20	-0.03
PRY	0.03	-0.02	0.08
COL	0.04	-0.04	0.11
MEX	0.04	-0.02	0.11
DOM	0.05	0.00	0.11
ECU	0.08	0.02	0.14
CUB	0.13	0.09	0.17
BRA	0.32	0.26	0.38
CRI	0.34	0.27	0.40
ARG	0.57	0.52	0.61
URY	0.66	0.61	0.71

Taller

Porcentajes con valores plausibles

Estimación de niveles de logro

Código 1.8: Porcentajes de estudiantes sobre el mínimo de competencia de logro en Lenguaje (Estudiantes 6to Grado, ERCE 2019)

Primero preparamos los datos (i.e., clustering, recodificación, y crear datos en TSL)

```
# -----  
# porcentajes con valores plausibles  
#-----  
  
#-----  
# cluster únicos  
#-----  
  
data_a6 <- data_a6 %>%  
erce::remove_labels() %>%  
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(.)))  
  
#-----  
# variable dummy para los niveles esperados  
#-----  
  
data_a6 <- data_a6 %>%  
  mutate(all = 1) %>%  
  mutate(lan_min_1 = case_when(  
    LAN_L1 == 'I' ~ 0,  
    LAN_L1 == 'II' ~ 0,  
    LAN_L1 == 'III' ~ 1,  
    LAN_L1 == 'IV' ~ 1)) %>%  
  mutate(lan_min_2 = case_when(  
    LAN_L2 == 'I' ~ 0,  
    LAN_L2 == 'II' ~ 0,  
    LAN_L2 == 'III' ~ 1,  
    LAN_L2 == 'IV' ~ 1)) %>%  
  mutate(lan_min_3 = case_when(  
    LAN_L3 == 'I' ~ 0,  
    LAN_L3 == 'II' ~ 0,  
    LAN_L3 == 'III' ~ 1,  
    LAN_L3 == 'IV' ~ 1)) %>%  
  mutate(lan_min_4 = case_when(  
    LAN_L4 == 'I' ~ 0,  
    LAN_L4 == 'II' ~ 0,  
    LAN_L4 == 'III' ~ 1,  
    LAN_L4 == 'IV' ~ 1)) %>%  
  mutate(lan_min_5 = case_when(  
    LAN_L5 == 'I' ~ 0,  
    LAN_L5 == 'II' ~ 0,  
    LAN_L5 == 'III' ~ 1,  
    LAN_L5 == 'IV' ~ 1)  
  )  
  
#-----  
# base de datos con diseño  
#-----  
  
erce_tsl <- survey::svydesign(  
  data = data_a6,  
  weights = ~WS,  
  strata = ~id_s,  
  id = ~id_j,  
  nest = TRUE)  
  
# Opción: corección a unidad primaria de muestreo que resulte  
# única al estrato  
  
library(survey)  
options(survey.lonely.psu="adjust")  
  
# Nota: withPV() solo funciona con Taylor Series linearization
```

El siguiente código es más complejo. En este empleamos la función `withPV()` de la librería `survey`. Esta función solo funciona con objetos TSL. En `mapping` indicamos los valores plausibles que emplearemos. En `design` se indica el objeto TSL. Luego en `action = function(erce_tsl)` escribimos la función de cálculo que emplearemos.

Finalmente, combinamos los resultados generados para cada valor plausible

```
# [...] código continua  
#-----  
# producir porcentajes con valores plausibles  
#-----  
  
options(scipen = 9999)  
options(digits = 5)  
  
tabla_7 <- withPV(  
  mapping = lan_min ~ lan_min_1 + lan_min_2 + lan_min_3 + lan_min_4 + lan_min_5,  
  data = erce_tsl,  
  action = quote(  
    svyby(~lan_min,  
    by = ~COUNTRY,  
    design = erce_tsl,  
    FUN = svymean  
  )  
)  
  
#-----  
# combinar resultados  
#-----  
  
library(mitools)  
summary(mitools::MIcombine(tabla_7))  
  
Multiple imputation results:  
  function(mapping, design, action, ...) UseMethod("withPV", design)  
  MIcombine.default(tabla_7)  
  results   se (lower upper) missInfo  
  ARG 0.31862 0.0120848 0.29493 0.34232 4 %  
  BRA 0.43514 0.0164518 0.40271 0.46756 14 %  
  COL 0.37490 0.0174680 0.34065 0.40916 4 %  
  CRI 0.54052 0.0175564 0.50570 0.57534 21 %  
  CUB 0.44568 0.0123999 0.42135 0.47001 6 %  
  DOM 0.16388 0.0109618 0.14237 0.18535 3 %  
  ECU 0.26106 0.0100840 0.24127 0.28086 7 %  
  GTM 0.15938 0.0101114 0.13954 0.17921 6 %  
  HND 0.16188 0.0148347 0.13277 0.19099 6 %  
  MEX 0.41612 0.0139756 0.38852 0.44371 16 %  
  NIC 0.12971 0.0083517 0.11318 0.14624 19 %  
  PAN 0.17483 0.0098372 0.15549 0.19417 11 %  
  PER 0.48984 0.0139409 0.46254 0.51724 5 %  
  PRY 0.18757 0.0105858 0.16674 0.20840 12 %  
  SLV 0.29335 0.0124945 0.26867 0.31802 17 %  
  URY 0.43723 0.0133617 0.41084 0.46362 17 %
```

Nota: este método tiene la desventaja de que los errores alrededor de los puntos estimados no están generados como errores de proporciones. No obstante, los puntos estimados son equivalentes a los reportados. Es posible generar los intervalos de confianza correctos, empleando la función `ldown::MIsvyciprop`. Sin embargo, los resultados se deben calcular por país. En el siguiente código ejemplificamos este ejercicio.

Código 1.9: Porcentajes de estudiantes sobre el mínimo de competencia de logo en Lenguaje (BRR) (Estudiantes 6to Grado, ERCE 2019)

Primero preparamos los datos (i.e., clustering, filtrado por país, recodificación, y crear listado de datos imputados).

```
# -----  
# porcentajes con valores plausibles con BRR  
#-----  
  
# cluster únicos y recodificaciones  
#-----  
  
arg_a6 <- data_a6 %>%  
  erce:remove_labels() %>%  
  dplyr::filter(COUNTRY == 'ARG') %>%  
  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(all = 1) %>%  
  mutate(lan_min_1 = case_when(  
    LAN_L1 == 'I' ~ 0,  
    LAN_L1 == 'II' ~ 0,  
    LAN_L1 == 'III' ~ 1,  
    LAN_L1 == 'IV' ~ 1)) %>%  
  mutate(lan_min_2 = case_when(  
    LAN_L2 == 'I' ~ 0,  
    LAN_L2 == 'II' ~ 0,  
    LAN_L2 == 'III' ~ 1,  
    LAN_L2 == 'IV' ~ 1)) %>%  
  mutate(lan_min_3 = case_when(  
    LAN_L3 == 'I' ~ 0,  
    LAN_L3 == 'II' ~ 0,  
    LAN_L3 == 'III' ~ 1,  
    LAN_L3 == 'IV' ~ 1)) %>%  
  mutate(lan_min_4 = case_when(  
    LAN_L4 == 'I' ~ 0,  
    LAN_L4 == 'II' ~ 0,  
    LAN_L4 == 'III' ~ 1,  
    LAN_L4 == 'IV' ~ 1)) %>%  
  mutate(lan_min_5 = case_when(  
    LAN_L5 == 'I' ~ 0,  
    LAN_L5 == 'II' ~ 0,  
    LAN_L5 == 'III' ~ 1,  
    LAN_L5 == 'IV' ~ 1)  
  )  
  
#-----  
# distinguir entre valores plausibles y otras variables  
#-----  
  
plausible_values <- dplyr::select(arg_a6,  
  lan_min_1, lan_min_2, lan_min_3, lan_min_4, lan_min_5)  
  
non_plausible_values <- dplyr::select(arg_a6, -one_of(names(plausible_values)))
```

Nota: para aplicar calcular los porcentajes de valores plausibles, requerimos instalar la librería **lodown**. Esto lo podemos realizar con el código **remotes::install_github("ajdamico/lodown")**, luego de tener instalada la librería **remotes**. Esta puede ser instalada con el comando: **install.packages("remotes")**.

```
# [...] código continua  
# -----  
# crear bases de datos por valor plausible  
# -----  
  
data_1 <- non_plausible_values %>%  
  dplyr::left_join(., dplyr::select(arg_a6, id_i, lan_min_1), by = 'id_i') %>%  
  rename(lan_min = lan_min_1)  
  
data_2 <- non_plausible_values %>%  
  dplyr::left_join(., dplyr::select(arg_a6, id_i, lan_min_2), by = 'id_i') %>%  
  rename(lan_min = lan_min_2)  
  
data_3 <- non_plausible_values %>%  
  dplyr::left_join(., dplyr::select(arg_a6, id_i, lan_min_3), by = 'id_i') %>%  
  rename(lan_min = lan_min_3)  
  
data_4 <- non_plausible_values %>%  
  dplyr::left_join(., dplyr::select(arg_a6, id_i, lan_min_4), by = 'id_i') %>%  
  rename(lan_min = lan_min_4)  
  
data_5 <- non_plausible_values %>%  
  dplyr::left_join(., dplyr::select(arg_a6, id_i, lan_min_5), by = 'id_i') %>%  
  rename(lan_min = lan_min_5)  
  
#-----  
# crear lista de datos imputados  
#-----  
  
arg_a6_imputed <- mitools::imputationList(  
  list(data_1, data_2, data_3, data_4, data_5)  
)  
  
#-----  
# crear objeto BRR con los datos imputados  
#-----  
  
library(survey)  
arg_brr <- survey::svrepdesign(  
  data = arg_a6_imputed,  
  type = ' Fay',  
  rho = .5,  
  weights = ~WS,  
  repweights = "repws[0-9]+",  
  combined.weights = TRUE  
)  
  
library(survey)  
options(survey.lonely.psu="adjust")  
  
#-----  
# producir porcentajes con valores plausibles  
#-----  
  
options(scipen = 999)  
options(digits = 9)  
  
tabla_8 <- lodown::MIsvyciprop(  
  ~lan_min,  
  design = arg_brr,  
  method = "Logit",  
  level = 0.95,  
  df = mean(unlist(lapply(arg_brr$designs,survey::degf)))  
)  
tabla_8
```

2.5%	97.5%
lan_min	0.31862 0.29606 0.3421

Notas: sobre la estimación de porcentajes de valores plausibles

- Cómo fuera indicado anteriormente, el cálculo de errores de porcentajes es sensible al método de cálculo de errores estándar.
- La función **survey::withPV()** facilita el cálculo de descriptivos que involucran a valores plausibles.
 - Pero no incluye el cálculo de proporciones con variables imputadas.
 - Y solo puede operar con objetos TSL.
- Por su parte la función **lodown::MIsvyciprop()** si permite el cálculo de proporciones basadas en valores plausibles
 - Los datos imputados pueden ser objetos TSL y BRR.
 - Y brinda errores estándar basados en el método "logit".
- Para estos escenarios el usuario secundario debe juzgar qué método es el más conveniente a sus propósitos. Si solo emplearan puntos estimados, el método más simple sería suficiente. Pero si las medidas de incertidumbre son parte del interés, entonces se debe elegir e implementar el método que entregue suficiente precisión para el propósito perseguido.
- En última instancia, es siempre quien produce los resultados quien es responsable por la fidelidad de los resultados generados.

```
#-----  
# resultados generados con withPV()  
#-----  
  
Multiple imputation results:  
function(mapping, design, action, ...) UseMethod("withPV",design)  
  MIcombine.default(tabla_7)  
    results (lower upper) missInfo  
  ARG   0.31862 0.29493 0.34232   4 %  
  
#-----  
# resultados generados con lodown::MIsvyciprop()  
#-----  
  
          2.5% 97.5%  
lan_min 0.31862 0.29606 0.3421  
  
#-----  
# comparación uno a uno  
#-----  
  
      estimate lower upper  
survey::withPV() 0.31862 0.29493 0.34232  
lodown::MIsvyciprop() 0.31862 0.29606 0.3421  
  
# Nota: diferencias al tercer decimal en los errores estimados.
```

Taller

Percentiles con valores plausibles

Estimación de percentiles empleando valores plausibles

Código 1.10: percentiles con valores plausibles (BRR)

- Para este ejemplo emplearemos los valores plausibles de los puntajes de Lenguaje. Los pasos empleados para calcular estos descriptivos son los siguientes:
 - cargar los datos
 - crear variables únicas de clustering
 - crear lista de datos imputados
 - crear un objeto BRR, con la lista de bases de datos imputadas
 - aplicar las funciones que calculan percentiles empleando diseño
 - editar tabla de resultados para que sea más legible
 - desplegar la tabla

```
# -----#
# percentiles con valores plausibles
# -----#
#-----#
# cluster únicos
#-----#
data_example <- data_a6 %>%
    erce::remove_labels() %>%
    mutate(id_j = as.numeric(as.factor(paste0(COUNTRY, " ", IDSCHOOL)))) %>%
    mutate(id_i = seq(1:nrow(.)))
#-----#
# separar valores plausibles
#-----#
plausible_values <- dplyr::select(data_example, LAN_1, LAN_2, LAN_3, LAN_4, LAN_5)
non_plausible_values <- dplyr::select(data_example, -one_of(names(plausible_values)))
```

```
# -----
# crear lista de datos imputados
# -----
data_1 <- non_plausible_values %>%
    dplyr::left_join(., dplyr::select(data_example, id_i, LAN_1), by = 'id_i') %>%
    rename(lan = LAN_1)

data_2 <- non_plausible_values %>%
    dplyr::left_join(., dplyr::select(data_example, id_i, LAN_2), by = 'id_i') %>%
    rename(lan = LAN_2)

data_3 <- non_plausible_values %>%
    dplyr::left_join(., dplyr::select(data_example, id_i, LAN_3), by = 'id_i') %>%
    rename(lan = LAN_3)

data_4 <- non_plausible_values %>%
    dplyr::left_join(., dplyr::select(data_example, id_i, LAN_4), by = 'id_i') %>%
    rename(lan = LAN_4)

data_5 <- non_plausible_values %>%
    dplyr::left_join(., dplyr::select(data_example, id_i, LAN_5), by = 'id_i') %>%
    rename(lan = LAN_5)

data_imputed <- mitools::imputationList(
    list(data_1, data_2, data_3, data_4, data_5)
)
#-----
# crear objeto BRR con los datos imputados
#-----

library(survey)
erce_brr_imp <- survey::svrepdesign(
    data = data_imputed,
    type = " Fay",
    rho = .5,
    weights = ~WS,
    repweights = "repws[0-9]+",
    combined.weights = TRUE
)
#-----
# calcular percentiles
#-----
percentiles_imp <- mitools::MCcombine(with(erce_brr_imp,
    svyquantile(~lan, design=erce_brr_imp,
    quantile=c(.05, .10, .25, .50, .75, .90, .95)
)))
#-----
# editar table
#-----
tabla_10 <- summary(percentiles_imp) %>%
    tibble::rownames_to_column("percentiles") %>%
    dplyr::rename(e = results, , se = se) %>%
    dplyr::select(percentiles, e, se)
# -----
# desplegar tabla
# -----
knitr::kable(tabla_10, digits = 2)
```

percentiles	e	se
[lan.0.05	519.06	2.64
[lan.0.1	554.68	2.27
[lan.0.25	617.60	1.75
[lan.0.5	694.70	1.39
[lan.0.75	775.14	1.16
[lan.0.9	842.08	1.25
[lan.0.95	879.16	1.25

Taller

Medias de valores plausibles

Estimación de medias empleando valores plausibles

Código 1.11: Medias de Lenguaje empleando valores plausibles por país (Estudiantes 6to Grado, ERCE 2019)

Preparamos los datos (i.e., clustering, crear datos en TSL) y luego, calculamos las medias por país, al interior de la función `withPV`.

```
# -----#
# porcentajes con valores plausibles
# -----#

# -----#
# cluster únicos
# -----#

data_a6 <- data_a6 %>%
  erce:::remove_labels() %>%
  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
#-----#

erce_tsl <- survey::svydesign(
  data      = data_a6,
  weights   = ~WS,
  strata    = ~id_s,
  id        = ~id_j,
  nest      = TRUE)

# Opción: corección a una unidad primaria de muestreo que resulte
# única al estrato

library(survey)
options(survey.lonely.psu="adjust")

# Nota: withPV() solo funciona con Taylor Series linearization
#-----#
# producir porcentajes con valores plausibles
#-----#

options(scipen = 999)
options(digits = 5)

tabla_11 <- withPV(
  mapping = lan ~ LAN_1 + LAN_2 + LAN_3 + LAN_4 + LAN_5,
  data    = erce_tsl,
  action   = quote(
    svyby(~lan,
          by = ~COUNTRY,
          FUN = svymean,
          design = erce_tsl
    )
  )
#-----#
# combinar resultados
#-----#

library(mitoools)
summary(mitoools::MIcombine(tabla_11))
```

Con el código anterior obtenemos los siguientes resultados.

```
Multiple imputation results:
  withPV.survey.design(mapping = lan ~ LAN_1 + LAN_2 + LAN_3 +
  LAN_4 + LAN_5, data = erce_tsl, action = quote(svyby(~lan,
  by = ~COUNTRY, FUN = svymean, design = erce_tsl)))
  MIcombine.default(tabla_11)
results  se (lower upper) missInfo
ARG 697.64 3.7325 690.26 705.02 18 %
BRA 734.44 3.7340 727.12 741.77 4 %
COL 719.32 4.4446 710.60 728.03 4 %
CRI 757.26 3.8698 749.66 764.86 8 %
CUB 737.93 2.9155 732.21 743.66 7 %
DOM 643.96 4.0316 636.03 651.89 12 %
ECU 684.26 3.0696 678.21 690.30 12 %
GTM 645.28 4.2125 636.98 653.57 13 %
HND 661.26 4.5049 652.41 670.11 9 %
MEX 725.56 3.4534 718.75 732.37 15 %
NIC 654.15 3.0884 648.07 660.23 13 %
PAN 651.62 3.8491 644.07 659.17 7 %
PER 741.23 4.1043 733.18 749.28 5 %
PRY 657.17 3.3849 650.48 663.86 17 %
SLV 698.77 3.2176 692.43 705.11 14 %
URY 734.06 3.5249 727.14 740.98 7 %
```

Podemos editar los resultados obtenidos para que sean más amigables.

```
#-----#
# editar tabla
#-----#

estimados <- summary(mitoools::MIcombine(tabla_11))

tabla_11 <- estimados %>%
  tibble::rownames_to_column("países") %>%
  rename(
    lan = results,
    lan_se = se,
    ll = 4,
    ul = 5,
    miss = 6
  )
#-----#
# mostrar tabla
#-----#

options(digits=10)
options(scipen = 999999)
knitr::kable(tabla_11, digits = 2)
```

países	lan	lan_se	ll	ul	miss
ARG	697.64	3.73	690.26	705.02	18 %
BRA	734.44	3.73	727.12	741.77	4 %
COL	719.32	4.44	710.60	728.03	4 %
CRI	757.26	3.87	749.66	764.86	8 %
CUB	737.93	2.92	732.21	743.66	7 %
DOM	643.96	4.03	636.03	651.89	12 %
ECU	684.26	3.07	678.21	690.30	12 %
GTM	645.28	4.21	636.98	653.57	13 %
HND	661.26	4.50	652.41	670.11	9 %
MEX	725.56	3.45	718.75	732.37	15 %
NIC	654.15	3.09	648.07	660.23	13 %
PAN	651.62	3.85	644.07	659.17	7 %
PER	741.23	4.10	733.18	749.28	5 %
PRY	657.17	3.38	650.48	663.86	17 %
SLV	698.77	3.22	692.43	705.11	14 %
URY	734.06	3.52	727.14	740.98	7 %

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