

Análisis de datos como desarrollo

Regresiones

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

Regresiones

Estimación de relaciones entre variables

Regresiones

- Revisaremos dos formas de relaciones entre variables:
 - Regresiones **lineales**
 - En estos modelos especificamos una variable de respuesta continua, con otras covariables (e.g., continuas o categóricas)
 - Regresiones **logísticas**
 - En estos modelos especificamos una variable de respuesta dicotómica, con otras covariables (e.g., continuas o categóricas)
- Desarrollaremos ejemplos con este tipo de modelos para escenarios que incluyen:
 - a variables representadas por **valores plausibles**
 - a variables observadas

Regresiones

- Revisaremos dos formas de relaciones entre variables:
 - Regresiones **lineales**
 - En estos modelos especificamos una variable de respuesta continua, con otras covariables (e.g., continuas o categóricas)
 - Regresiones **logísticas**
 - En estos modelos especificamos una variable de respuesta dicotómica, con otras covariables (e.g., continuas o categóricas)
- Desarrollaremos ejemplos con este tipo de modelos para escenarios que incluyen:
 - a variables representadas por **valores plausibles**
 - a variables observadas

Para el caso de atributos de estudiantes, en estudios de gran escala, podemos formalizar estos modelos de la siguiente forma:

$$y_{ij} = \alpha + \beta x_{ij} + \delta v_{.j} + \epsilon_{ij}$$

Donde, y_{ij} es nuestra variable de interés, comúnmente los puntajes de las pruebas, u otro puntaje continuo que representa a un atributo de los estudiantes, y un conjunto de otros factores reportados por los estudiantes (x_{ij}), u otros factores escuela ($v_{.j}$), reportados por otros actores (e.g., información administrativa, respuesta de los directores, o de los profesores).

En el caso de la regresión logística, empleamos una ecuación similar:

$$\text{logit}\{Pr(y_{ij=1})\} = \alpha + \beta x_{ij} + \delta v_{.j}$$

En este caso, estamos interesados en cómo varía una variable de respuesta dicotómica, condicional a otras variables que nos aportan los estudiantes (x_{ij}), u otros factores escuela ($v_{.j}$) que provienen de otras fuentes.

... A continuación, veremos diferentes ejemplos de aplicaciones de estos modelos, sobre los datos de ERCE 2019.

Taller

Preparación del entorno R

Librerías que emplearemos en esta sesión

Librerías en uso

- A la derecha de la presente lámina, incluimos el conjunto de paquetes de R que estaremos empleando.
- En particular, la librería **library(erce)**, la cual nos da acceso a los datos que serán liberados y que nos permiten replicar los resultados generados para los informes.
- Además en este librería se incluyen unas pocas funciones auxiliares para operar con los datos de ERCE
 - **remove_labels()** que la necesitamos para remover toda meta-data de los datos que estamos empleando
 - **combine_reg()** ocuparemos esta función para producir los output de resultados de regresiones lineales
 - **combine_log()** ocuparemos esta función para producir los output de resultados de regresiones logísticas

Librerías en uso

- A la derecha de la presente lámina, incluimos el conjunto de paquetes de R que estaremos empleando.
- En particular, la librería **library(erce)**, la cual nos da acceso a los datos que serán liberados y que nos permiten replicar los resultados generados para los informes.
- Además en este librería se incluyen unas pocas funciones auxiliares para operar con los datos de ERCE
 - **remove_labels()** que la necesitamos para remover toda meta-data de los datos que estamos empleando
 - **combine_reg()** ocuparemos esta función para producir los output de resultados de regresiones lineales
 - **combine_log()** ocuparemos esta función para producir los output de resultados de regresiones logísticas

```
# -----
# librerias
# -----
# Nota: instalar librerias manualmente.

#-----
# librerias para instalar librerias remotas
#-----

install.packages('devtools')
install.packages('remotes')
install.packages('credentials')

#-----
# libreria ERCE con datos
#-----

# needed for some computers with git installed
credentials::set_github_pat()

# library is in development and in private mode
devtools::install_github(
  'dacarras/erce',
  auth_token = 'ghp_0qXfVqkIi4AAZeV984H0GieflB45IN33iIEX',
  force = TRUE)

#-----
# librerías en uso
#-----

install.packages('tidyverse')
install.packages('mitools')
install.packages('survey')
install.packages('srvyr')
```

Taller

Preparación de datos

Agregando BRR normalizados

Código 1.1: pesos BRR con pesos estandarizados o pesos SENATE

- Al igual que en la sesión anterior, vamos a crear pesos BRR con peso normalizada tipo senate.
- Estos pesos creados nos permite ajustar modelos para más de un país en caso de que fuera necesario, gracias a que los pesos en esta escala permiten que las observaciones de cada país sean expandidas a una cifra equivalente. En este caso, a un total de 1000 casos.
- Estamos realizando esta operación mediante las líneas
 - `mutate(repws001 = BRR1/WT * WS) %>%`
 - `[...]`
 - `mutate(repws100 = BRR1/WT * WS) %>%`
 - para cada uno de los pesos replicados.
- Con los nuevos pesos replicados generados (`repws001`-`repws100`), podemos implementar diferentes análisis que emplean BRR, pero ahora incluyendo pesos senate o pesos normalizados.
- Este paso previo es importante para la generación de resultados regionales, o aplicados sobre la "*pooled sample*".
- Esto nos permite producir cálculos de porcentajes, medias, y regresiones , donde cada país pesa de forma equivalente.

Código 1.1: pesos BRR con pesos estandarizados o pesos SENATE

- Al igual que en la sesión anterior, vamos a crear pesos BRR con peso normalizada tipo senate.
- Estos pesos creados nos permite ajustar modelos para más de un país en caso de que fuera necesario, gracias a que los pesos en esta escala permiten que las observaciones de cada país sean expandidas a una cifra equivalente. En este caso, a un total de 1000 casos.
- Estamos realizando esta operación mediante las líneas
 - `mutate(repws001 = BRR1/WT * WS) %>%`
 - [...]
 - `mutate(repws100 = BRR1/WT * WS) %>%`
 - para cada uno de los pesos replicados.
- Con los nuevos pesos replicados generados (`repws001`-`repws100`), podemos implementar diferentes análisis que emplean BRR, pero ahora incluyendo pesos senate o pesos normalizados.
- Este paso previo es importante para la generación de resultados regionales, o aplicados sobre la "pooled sample".
- Esto nos permite producir cálculos de porcentajes, medias, y regresiones , donde cada país pesa de forma equivalente.

```
# -----  
# BRR con pesos senate  
# -----  
  
# -----  
# BRR/WT factor del BRR sobre los observados  
# -----  
  
data_a6 <- erce::erce_2019_qa6 %>%  
erce$remove_labels() %>%  
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

Rregresiones

Ajustando regresiones con valores plausibles y Taylor Series Linearization (TSL)

Código 1.2: regresión de puntajes de matemáticas a nivel socioeconómico, empleando datos de la región (Estudiantes 6to grado, ERCE 2019)

- En este ejemplo, empleamos los puntajes de matemáticas. Al igual que en rutinas previas, primero vamos a crear las variables de *clustering*, de modo que en un mismo modelo, escuelas y estratos sean únicos entre todos los países (i.e., que no se repita el mismo código).
- Luego, creamos el objeto `erce_tsl`, el cual es la base de datos con las características de diseño que necesitamos para realizar las estimaciones.
- Despues, ajustamos el modelo `mat ~ 1 + ISECF`, el cual representa al modelo formal $y_{ij} = \alpha + \beta s_{es} + \epsilon_{ij}$
- Finalmente, inspeccionamos el output del modelo con `summary(mitoools::MIcombine())`
- En términos generales la región presenta un promedio de 697.56 puntos a niveles medios de nivel socioeconómico ($\alpha = 697.56$, $SE = 1.00$, $CI95\% [695.54; 699.60]$).
- Y por otro lado, la asociación general con las diferencias en nivel socioeconómico, entre los estudiantes de sexto de la región es de 36.15 puntos ($\beta = 36.15$, $SE = 0.79$, $CI95\% [34.60; 37.70]$). Es decir, los estudiantes a una desviación estándar de más nivel socioeconómico que el resto de sus pares, esperamos observar puntajes más altos en un 36.15 puntos (lo que corresponde a 35% de desviación estándar de los puntajes de matemáticas).
- Este último término 35% de desviación estándar de los puntajes de matemáticas, lo obtenemos mediante la división de β por la desviación estándar de la región, de los puntajes de matemáticas (100 puntos).

```

# # regresion con valores plausibles y TSL
# -----
#-
# 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: corrección 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
#--#
# ajustar regresión
#--#

options(scipen = 999)
options(digits = 10)

tabla_1 <- mitools::withPV(
  mapping = mat ~ MAT_1 + MAT_2 + MAT_3 + MAT_4 + MAT_5,
  data   = erce_tsl,
  action  = quote(survey::svyglm(mat ~ 1 + ISECF, design = erce_tsl)),
  rewrite=TRUE
)

#--#
# combinar resultados
#--#

library(mitoools)
summary(mitoools::MIcombine(tabla_1))

Multiple imputation results:
  withPV.survey.design(mapping = mat ~ MAT_1 + MAT_2 + MAT_3 +
    MAT_4 + MAT_5, data = erce_tsl, action = quote(survey::svyglm(mat ~
    1 + ISECF, design = erce_tsl)), rewrite = TRUE)
  MIcombine.default(tabla_1)
  results          se      (lower       upper) missInfo
  (Intercept) 697.5691108 1.0017864349 695.54088295 699.59733857 36 %
  ISECF        36.1518283 0.7905758005 34.59531868 37.70833792 13 %

```

Taller

Rregresiones

Regresiones con Taylor Series Linearization (TSL) incluyendo valores p

Código 1.3: regresión con valores plausibles y valores *p*.

- En el ejemplo anterior la combinación de estimados por cada valor, no nos entrega los valores *p* de los estimados obtenidos.
 - Por otro lado, si tenemos los intervalos de confianza de los estimados podemos llegar a las mismas conclusiones sustantivas.
 - Si los intervalos de confianza de un estimado incluyen al valor cero, podemos concluir que hemos obtenido estimados que son típicos a lo esperado por la hipótesis nula.
 - Por el contrario, si los intervalos de confianza no incluyen al valor cero, entonces tenemos evidencias de que nuestros estimados no son típicos a la distribución de la hipótesis nula.
- No obstante, es posible que **sea un requisito** reportar valores *p* de los estimados generados.
 - Para poder obtener los valores *p* de los estimados obtenidos necesitamos aplicar una prueba "t", empleando la información que tenemos del tamaño del estimado, su error estándar, y los grados de libertad del modelo ajustado.
 - Vamos a emplear una función auxiliar, incluida en la librería **erce**, para incluir esta información, que no está siendo entregada por **summary(mitoools::MIcombine(tabla_1))**.
 - **erce::combine_reg()**
 - **erce::combine_reg()** nos entrega la misma información que obtuvimos anteriormente, pero adicionalmente agrega una columna al output con el valor *p* de los estimados (**p_val**).

Nota: estos resultados son equivalentes a los obtenidos por Mplus, con el mismo método (TSL) tanto en los puntos estimados, como en los errores estándar.

```
# # regresion con valores plausibles y TSL
# -----
#-
#- # 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: corrección 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

#-- # ajustar modelo
#--

pv_estimates <- mitools::withPV(
  mapping = mat ~ MAT_1 + MAT_2 + MAT_3 + MAT_4 + MAT_5,
  data   = erce_tsl,
  action  = quote(survey::svyglm(mat ~ 1 + ISECF, design = erce_tsl)),
  rewrite=TRUE
)

#-- # combinar resultados
#--

tabla_2 <- erce::combine_reg(pv_estimates)

#-- # mostrar tabla
#--

tabla_2 %>%
  knitr::kable(., digits = 2)
```

term	e	se	p_val	lo	hi
(Intercept)	697.57	1.00	0	695.54	699.60
ISECF	36.15	0.79	0	34.60	37.71

Taller

Rregresiones

Regresiones empleando Balanced Repeated Replication (BRR)

Código 1.4: regresión con BRR empleando **survey**

- En el siguiente ejemplo, empleamos la librería **survey**
 - Pero en esta variante estamos empleando el método de BRR
 - Este método nos va entregar los mismos puntos estimados, pero con errores estándar más pequeños que en el caso de TSL
- Los pasos que estamos empleando para producir los resultados, son los mismos que los anteriores
 - Adaptamos los datos originales a lo que necesitamos: (e.g., creamos cluster únicos, creamos un vector común para los valores plausibles).
 - Creamos un objeto BRR, empleando una lista de datos imputados
 - Ajustamos un modelo de regresión dentro de la función **with()**
 - combinamos los estimados y producimos la tabla de resultados

Nota: estos resultados son equivalentes a los obtenidos por Mplus, con el mismo método (BRR) tanto en los puntos estimados, como en los errores estándar.

```
# -----  
# regresion con valores plausibles y TSL  
# -----  
  
#-----  
# 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(.)))
```

```
# [...]  
#-----  
# lista de datos imputados  
#-----  
  
data_1 <- data_a6 %>% mutate(matpv = MAT_1)  
data_2 <- data_a6 %>% mutate(matpv = MAT_2)  
data_3 <- data_a6 %>% mutate(matpv = MAT_3)  
data_4 <- data_a6 %>% mutate(matpv = MAT_4)  
data_5 <- data_a6 %>% mutate(matpv = MAT_5)  
  
data_imputed <- mitools::imputationList(  
  list(data_1, data_2, data_3, data_4, data_5))  
  
#-----  
# base de datos con diseño  
#-----  
  
erce_brr <- survey::svrepdesign(  
  data = data_imputed,  
  type = 'Fay',  
  rho = .5,  
  weights = ~WS,  
  repweights = "repws[0-9]+",  
  combined.weights = TRUE  
)  
  
# Opción: corección a unidad primaria de muestreo que resulte  
# única al estrato  
  
library(survey)  
options(survey.lonely.psu="adjust")  
  
#-----  
# especificar model  
#-----  
  
regression_model <- as.formula(matpv ~ 1 + ISECF)  
  
#-----  
# ajustar modelo  
#-----  
  
svyglm_estimates <- with(erce_brr,  
  survey::svyglm(regression_model, design = erce_brr  
)  
  
#-----  
# combinar estimados  
#-----  
  
tabla_3 <- erce::combine_reg(svyglm_estimates)  
  
#-----  
# mostrar resultados  
#-----  
  
knitr::kable(tabla_3, digits = 2)
```

term	e	se	p_val	lo	hi
(Intercept)	697.57	0.95	0	695.62	699.52
ISECF	36.15	0.82	0	34.54	37.76

Taller

Regresión Logística

Regresión Logística empleando Taylor Series Linearization (TSL) y valores plausibles

Código 1.5: regresión con TSL empleando **survey** prediciendo una variable dicotómica

- Los valores plausibles de ERCE 2019, se encuentran en la base de datos como *niveles de logro*
 - Estos valores son variables categóricas ordinales, que incluyen a los valores 'I', 'II', 'III', y 'IV'. A mayor valor ordinal, mayor logro obtenido en la habilidad evaluada. Para el caso de Matemáticas, estos valores se encuentran en las variables **MAT_L1-MAT_L5**. Los niveles III y IV son los valores de logro mínimo esperado (*minimum proficiency level = MPL*). Los estudiantes que alcanzan los MPL en matemáticas, son los que (UNESCO, 2021 p18):

... capaces de resolver problemas que requieren **interpretar información** en diversos formatos incluyendo **tablas y gráficos**; recurrir a **dos o más operaciones aritméticas**; estimar áreas y perímetro; calcular adiciones y sustracciones de fracciones; e identificar relaciones de perpendicularidad y paralelismo en el plano.

- En este ejemplo, emplearemos estas variables ordinales, para crear una variable dicotómica que exprese el MPL de interés, y la condicionaremos con otras variables.

```
# -----  
# regresión logística con valores plausibles y TSL  
# -----  
  
# -----  
# 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_l = seq(1:nrow(.)))  
  
# -----  
# elegimos un país al azar  
# -----  
  
data_a6 <- data_a6 %>%  
  dplyr::filter(IDCNTRY == 76) # Brasil
```

```
#--  
# recodificación  
#-----  
  
rec_pv_level <- function(x){  
  dplyr::case_when(  
    x == 'I' ~ 0,  
    x == 'II' ~ 0,  
    x == 'III' ~ 1,  
    x == 'IV' ~ 1)  
}  
  
data_a6 <- data_a6 %>%  
  mutate(mat_mpl1 = rec_pv_level(MAT_L1)) %>%  
  mutate(mat_mpl2 = rec_pv_level(MAT_L2)) %>%  
  mutate(mat_mpl3 = rec_pv_level(MAT_L3)) %>%  
  mutate(mat_mpl4 = rec_pv_level(MAT_L4)) %>%  
  mutate(mat_mpl5 = rec_pv_level(MAT_L5))  
  
#--  
# 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")  
  
#--  
# ajustar modelo  
#-----  
  
logit_estimates <- mitools::withPV(  
  mapping = mat_mpl ~ mat_mpl1 + mat_mpl2 + mat_mpl3 + mat_mpl4 + mat_mpl5,  
  data = erce_tsl,  
  action = quote(  
    survey:::svyglm(mat_mpl ~ 1 + EDU,  
      design = erce_tsl,  
      family = 'quasibinomial'(link = "logit")  
    )),  
  rewrite=TRUE  
)  
  
#--  
# combinar estimados  
#-----  
  
tabla_4 <- combine_log(logit_estimates)  
  
#--  
# mostrar resultados  
#-----  
  
knitr::kable(tabla_4, digits = 2)
```

term	e	se	p_val	lo	hi	or	or_lo	or_hi
(Intercept)	-1.20	0.08	0	-1.36	-1.04	0.30	0.26	0.35
EDU	1.53	0.14	0	1.26	1.80	4.61	3.52	6.04

Taller

Regresión Logística

Regresión Logística empleando Balanced Repeated Replication (BRR) y valores plausibles

Código 1.6: regresión con BRR empleando **survey** prediciendo una variable dicotómica

- Para este ejemplo, emplearemos las mismas variables de respuesta anteriores.
 - Los niveles III y IV son los valores de logro mínimo esperado (*minimum proficiency level = MPL*).
- En este ejemplo, emplearemos estas variables ordinales, para crear una variable dicotómica que exprese el MPL de interés, y la condicionaremos con otras variables

Nota: estos resultados no pueden ser generados con Mplus.

```
#-- regresión logística con valores plausibles y BRR
#-----#
# 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(.)))
#
# elegimos un país al azar
#
data_a6 <- data_a6 %>%
  dplyr::filter(IDCNTRY == 76) # Brasil
#
# recodificación
#
rec_pv_level <- function(x){
  dplyr::case_when(
    x == 'I' ~ 0,
    x == 'II' ~ 0,
    x == 'III' ~ 1,
    x == 'IV' ~ 1)
}
```

```
#
# lista de datos imputados
#-----
data_1 <- data_a6 %>% mutate(mat_mpl = rec_pv_level(MAT_L1))
data_2 <- data_a6 %>% mutate(mat_mpl = rec_pv_level(MAT_L2))
data_3 <- data_a6 %>% mutate(mat_mpl = rec_pv_level(MAT_L3))
data_4 <- data_a6 %>% mutate(mat_mpl = rec_pv_level(MAT_L4))
data_5 <- data_a6 %>% mutate(mat_mpl = rec_pv_level(MAT_L5))

data_imputed <- mitools::imputationList(
  list(data_1, data_2, data_3, data_4, data_5)
)
#
# base de datos con diseño
#-----
erce_brr <- survey::svrepdesign(
  data = data_imputed,
  type = 'Fay',
  rho = .5,
  weights = ~WS,
  repweights = "repws[0-9]+",
  combined.weights = TRUE
)
#
# Opción: corrección a unidad primaria de muestreo que resulte
# única al estrato
library(survey)
options(survey.lonely.psu="adjust")
#
# especificar modelo
#-----
logit_model <- as.formula(mat_mpl ~ 1 + EDU)

#
# ajustar modelo
#-----

svyglm_estimates <- with(erce_brr,
  survey::svyglm(logit_model,
  design = erce_brr,
  family = 'quasibinomial'(link = "logit"))
)
#
# combinar estimados
#-----
tabla_6 <- erce::combine_log(svyglm_estimates)

#
# mostrar resultados
#-----
knitr::kable(tabla_6, digits = 2)

# Nota: Mplus no puede ajustar un modelo de regresión logística con BRR,
# por defecto para este escenario MPLUS ajusta modelos probit.
```

term	e	se	p_val	lo	hi	or	or_lo	or_hi
(Intercept)	-1.20	0.08		-1.36	-1.04	0.30	0.26	0.35
EDU	1.53	0.14		1.26	1.80	4.61	3.52	6.04

Taller

Regresión Logística

Interpretación de resultados

Regresión logística

La regresión logística que ajustamos en el ejemplo anterior, la podemos representar de la siguiente manera:

$$\text{logit}\{Pr(y_{ij}=1)\} = \alpha + \beta \text{EDU}_{ij}$$

Donde, **EDU == 0** representa a los estudiantes donde sus padres alcanzan niveles de escolaridad menores a la educación terciaria; mientras que **EDU == 1** representa a los estudiantes donde, padre o madre en su familia ha alcanzado niveles de educación terciaria.

Nuestros estimados originalmente están en la escala *logit*. Y por tanto, los podemos interpretar en términos de como esperamos que cambia la probabilidad de que $Pr(y_{ij}=1)$, en la escala *logit*. Estos resultados nos indican que los estudiantes hijos de padres con educación terciaria tienen más chances de alcanzar los niveles de esperados de competencia matemática al grado (MPL) (logits = 1.53; SE = 0.14;p <.001).

Una forma alternativa de interpretar estos resultados, es convertir los estimados a **odds ratio** u **OR**. Los OR se obtienen como la "exponenciación" de los estimados logits ($\exp(\beta)$). Si interpretamos los **odds ratio** del modelo ajustado, podemos indicar que en términos generales los estudiantes de Brasil, hijos de familias donde padres o madres poseen educación terciaria, tienen más de 4.61 veces más chances, que el resto de sus compañeros de lograr los niveles mínimos de competencia matemática (logits = 1.53; SE = 0.14; OR = 4.61; p <.001; CI95% [3.52; 6.04]).

```
# regresión logistica con valores plausibles y BRR
# [...]
#--#
# ajustar modelo
#--#
svyglm_estimates <- with(erce_brr,
  survey:::svyglm(mat_mpl ~ 1 + EDU,
    design = erce_brr,
    family = 'quasibinomial'(link = "logit"))
)
#--#
# combinar estimados
#--#
tabla_6 <- ercombine_log(svyglm_estimates)
#--#
# mostrar resultados
#--#
knitr::kable(tabla_6, digits = 2)

| term | e | se | p_val | lo | hi | or | or_lo | or_hi |
|:-----|---:|---:|---:|---:|---:|---:|---:|---:|
|(Intercept) | -1.20 | 0.08 | 0 | -1.36 | -1.04 | 0.30 | 0.26 | 0.35 |
|EDU | 1.53 | 0.14 | 0 | 1.26 | 1.80 | 4.61 | 3.52 | 6.04 |
```

Nota: Mplus no puede ajustar un modelo de regresión logística con BRR,
por defecto para este escenario MPLUS ajusta modelos probit.

Taller

Regresión Lineal

Regresión Lineal con variables observadas, empleando TSL

Código 1.7: regresión variable observada, y dos predictores con TSL

- Para este ejemplo, empleamos los datos de Cuba (ERCE 2019, Sexto grado)
- En particular los datos de Autorregulación Escolar (AURES), y los condicionamos a los valores de nivel socioeconómico de las familias (ISECF), y a los valores del indicador de victimización del estudiante en la escuela (VIOES).
- Los resultados incluidos en el informe "Habilidades socioemocionales en América Latina y el Caribe", los valores de estas covariables fueron centrados de una forma particular, antes de ser ingresados al modelo.
- En general, el modelo ajustado era de la siguiente forma:

$$y_{ij} = \alpha + \beta(x_{ij} - \bar{x}_{..}) + \pi_w z_w + \pi_b z_b + \epsilon_{ij}$$

Donde, y_{ij} representa a los puntajes de AURES, x_{ij} representa a los puntajes de VIOES, y z_{ij} representa a los puntajajes de ISCEC, centrados a cada escuela, y al promedio de los niveles de cada escuela ($z_w = (z_{ij} - \bar{z}_{.j})$; $z_b = (\bar{z}_{.j} - \bar{z}_{..})$).

Esta forma de especificar el modelo de regresión es una adaptación del modelo desagregado de Rights et al., (2019), a un modelo poblacional. Lo distintivo de esta especificación, es que los puntajes de nivel socioeconómico se encuentran descompuestos entre la variabilidad al interior de las escuelas (z_w), y la variabilidad entre las escuelas (z_b).

En este ejemplo, esta descomposición es informativa porque los resultados de $\pi_w = .46$ y $\pi_b = -1.74$ difieren de signo; y por tanto nos entregan información que no econtraríamos sin la descomposición de nivel socioeconómico ($z_{ij} = z_w + z_b$).

```
# # regresión lineal a variable observada con dos predictores
# 

#-----#
# preparacion de datos
#-----#


data_a6 <- erce::erce_2019_qa6 %>%
# removemos labels
erce::remove_labels() %>%
# elegimos un solo país
dplyr::filter(IDCNTRY == 192) %>% # Cuba
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(.))) %>%
# centrado de nivel socioeconómico
mutate(z = ISECF) %>% # mean score
mutate(z_c = erce::c_mean(z, id_j)) %>% # means by group
mutate(z_g = erce::c_mean(z, id_k)) %>% # grand mean
mutate(z_w = z - z_c) %>% # centering within cluster
mutate(z_m = z - z_g) %>% # centering to the grand mean
mutate(z_b = z_c - z_g) %>% # centered cluster means
## violencia en la escuela (índice de victimización escolar)
mutate(x = VIOES) %>% # mean score
mutate(x_c = erce::c_mean(x, id_j)) %>% # means by group
mutate(x_g = erce::c_mean(x, id_k)) %>% # grand mean
mutate(x_w = x - x_c) %>% # centering within cluster
mutate(x_m = x - x_g) %>% # centering to the grand mean
mutate(x_b = x_c - x_g) %>% # centered cluster means

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


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

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

#-----#
# ajustar modelo
#-----#


svy_estimates <- survey::svyglm(AURES ~ 1 + x_m + z_w + z_b, design = erce_tsl)

#-----#
# extraer estimados a una tabla
#-----#


tabla_7 <- broom::tidy(svy_estimates, conf.int = TRUE)

#-----#
# mostrar tabla
#-----#


knitr::kable(tabla_7, digits = 2)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	56.78	0.21	275.83	0.00	56.37	57.19
x_m	-3.18	0.20	-16.26	0.00	-3.57	-2.80
z_w	0.46	0.22	2.13	0.03	0.04	0.89
z_b	-1.74	0.56	-3.13	0.00	-2.83	-0.64

Taller

Regresión Lineal

Regresión Lineal con variables observadas, empleando BRR

Código 1.8: regresión variable observada, y dos predictores con BRR

- En el siguiente ejemplo replicamos la especificación anterior, pero empleando el método BRR para el cálculo de errores estándar.
- En resumen estamos ajustando una regresión sobre los datos de Autorregulación Escolar (AURES), y los condicionamos a los valores de nivel socioeconómico de las familias (ISECF), y a los valores del indicador de victimización del estudiante en la escuela (VIOES), sobre los datos de Cuba (ERCE 2019, Sexto Grado).
- Formalmente, estamos ajustando el siguiente modelo:

$$y_{ij} = \alpha + \beta(x_{ij} - \bar{x}_{..}) + \pi_w z_w + \pi_b z_b + \epsilon_{ij}$$

Donde, y_{ij} representa a los puntajes de AURES, x_{ij} representa a los puntajes de VIOES, y z_{ij} representa a los puntajes de ISCECF, centrados a cada escuela, y al promedio de los niveles de cada escuela ($z_w = (z_{ij} - \bar{z}_{..})$; $z_b = (\bar{z}_j - \bar{z}_{..})$). Los resultados obtenidos tanto por TSL y BRR son similares.

```
# -----#
# regresión lineal a variable observada con dos predictores
# -----
# preparacion de datos
# 

data_a6 <- erce::erce_2019_qa6 %>%
# removemos labels
erce::remove_labels() %>%
# elegimos un solo país
dplyr::filter(IDCTRY == 192) %>% # Cuba
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(.))) %>%
# centrado de nivel socioeconómico
mutate(z = ISECF) %>% # mean score
mutate(z_c = erce::c_mean(z, id_j)) %>% # means by group
mutate(z_g = erce::c_mean(z, id_k)) %>% # grand mean
mutate(z_w = z - z_c) %>% # centering within cluster
mutate(z_m = z - z_g) %>% # centering to the grand mean
mutate(z_b = z_c - z_g) %>% # centered cluster means
## violencia en la escuela (índice de victimización escolar)
mutate(x = VIOES) %>% # mean score
mutate(x_c = erce::c_mean(x, id_j)) %>% # means by group
mutate(x_g = erce::c_mean(x, id_k)) %>% # grand mean
mutate(x_w = x - x_c) %>% # centering within cluster
mutate(x_m = x - x_g) %>% # centering to the grand mean
mutate(x_b = x_c - x_g) # centered cluster means
```

```
# [...]
#-----#
# base de datos con diseño
#-----

erce_brr <- survey::svrepdesign(
  data    = data_a6,
  type    = ' Fay',
  rho     = .5,
  weights = ~WT,
  repweights = 'BRR[0-9]+',
  combined.weights = TRUE
)

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

#-----#
# ajustar modelo
#-----#

svy_estimates <- survey::svyglm(AURES ~ 1 + x_m + z_w + z_b, design = erce_brr)

#-----#
# extraer estimados a una tabla
#-----#

tabla_8 <- broom::tidy(svy_estimates, conf.int = TRUE)

#-----#
# mostrar tabla
#-----#

knitr::kable(tabla_8, digits = 2)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	56.78	0.22	259.54	0.00	56.35	57.21
x_m	-3.18	0.19	-16.93	0.00	-3.56	-2.81
z_w	0.46	0.23	1.99	0.05	0.00	0.93
z_b	-1.74	0.57	-3.03	0.00	-2.88	-0.60

Taller

Regresión logística

Regresión logística con variables observadas, empleando TSL

Código 1.9: regresión logística sobre variable observada, y dos predictores, empleando TSL

- En el siguiente ejemplo empleamos como variable de respuesta a REPC, variable que nos indica si los estudiantes han sido retenidos durante su trayectoria escolar. Esta variable toma valores 1, cuando los estudiantes han repetido al menos una vez, y toma valores 0, cuando los estudiantes no han repetido en ningún grado.
- Emplearemos los mismos predictores del ejemplo anterior: Victimización al interior de la escuela (VIOES), y nivel socioeconómico de las familias (ISECF)
- Ajustamos un modelo que es estructuralmente similar al modelo empleado en el ejemplo anterior.

$$\text{logit}\{Pr(y_{ij} = 1\} = \beta(x_{ij} - \bar{x}_{..}) + \pi_w z_w + \pi_b z_b$$

```
# -- regresión logística a variable observada con dos predictores
# --
# -- preparacion de datos
# --

data_a6 <- erce::erce_2019_qa6 %>%
  # removemos labels
  erce::remove_labels() %>%
  # elegimos un solo país
  dplyr::filter(IDCNTRY == 192) %>% # Cuba
  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(.))) %>%
  # centrado de nivel socioeconómico
  mutate(z = ISECF) %>% # mean score
  mutate(z_c = erce::c_mean(z, id_j)) %>% # means by group
  mutate(z_g = erce::c_mean(z, id_k)) %>% # grand mean
  mutate(z_w = z - z_c) %>% # centering within cluster
  mutate(z_m = z - z_g) %>% # centering to the grand mean
  mutate(z_b = z_c - z_g) %>% # centered cluster means
  ## violencia en la escuela (indice de victimización escolar)
  mutate(x = VIOES) %>% # mean score
  mutate(x_c = erce::c_mean(x, id_j)) %>% # means by group
  mutate(x_g = erce::c_mean(x, id_k)) %>% # grand mean
  mutate(x_w = x - x_c) %>% # centering within cluster
  mutate(x_m = x - x_g) %>% # centering to the grand mean
  mutate(x_b = x_c - x_g) %>% # centered cluster means
```

```
# [...]
#-----#
# base de datos con diseño
#-----

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

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

#--#
# ajustar modelo
#--#

svy_estimates <- survey::svyglm(REPC ~ 1 + x_m + z_w + z_b,
  design = erce_tsl,
  family = 'quasibinomial'(link = "logit")
)

#--#
# extraer estimados a una tabla
#--#

tabla_9 <- broom::tidy(svy_estimates, conf.int = TRUE) %>%
  mutate(or = exp(estimate)) %>%
  mutate(or_lo = exp(conf_low)) %>%
  mutate(or_hi = exp(conf_high))

#--#
# mostrar tabla
#--#

knitr::kable(tabla_9, digits = 2)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high	or	or_lo	or_hi
(Intercept)	-3.94	0.15	-27.02	0.00	-4.23	-3.66	0.02	0.01	0.03
x_m	0.53	0.10	5.53	0.00	0.34	0.72	1.70	1.41	2.06
z_w	-0.68	0.15	-4.37	0.00	-0.98	-0.37	0.51	0.37	0.69
z_b	0.33	0.28	1.18	0.24	-0.22	0.87	1.39	0.80	2.39

Taller

Regresión logística

Regresión logística con variables observadas, empleando BRR

Código 1.10: regresión logística sobre variable observada, y dos predictores, empleando BRR

- En el siguiente ejemplo empleamos como variable de respuesta a REPC, variable que nos indica si los estudiantes han sido retenidos durante su trayectoria escolar. Esta variable toma valores 1, cuando los estudiantes han repetido al menos una vez, y toma valores 0, cuando los estudiantes no han repetido en ningún grado.
- Emplearemos los mismos predictores del ejemplo anterior: Victimización al interior de la escuela (VIOES), y nivel socioeconómico de las familias (ISECF)
- Formalmente, estamos ajustando el siguiente modelo:

$$\text{logit}\{Pr(y_{ij} = 1\} = \beta(x_{ij} - \bar{x}_{..}) + \pi_w z_w + \pi_b z_b$$

La única diferencia con el mdoelo anterior, es que estamos empleando BRR para corregir los errores estándar.

```
# regresión lineal a variable observada con dos predictores
# ----
# preparación de datos
#
data_a6 <- erce::erce_2019_qa6 %>%
# removemos labels
erce::remove_labels() %>%
# elegimos un solo país
dplyr::filter(IDCTRY == 192) %% # Cuba
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(.))) %>%
# centrado de nivel socioeconómico
mutate(z = ISECF) %% # mean score
mutate(z_c = erce::c_mean(z, id_j)) %% # means by group
mutate(z_g = erce::c_mean(z, id_k)) %% # grand mean
mutate(z_w = z - z_c) %% # centering within cluster
mutate(z_m = z - z_g) %% # centering to the grand mean
mutate(z_b = z_c - z_g) %% # centered cluster means
## violencia en la escuela (índice de victimización escolar)
mutate(x = VIOES) %% # mean score
mutate(x_c = erce::c_mean(x, id_j)) %% # means by group
mutate(x_g = erce::c_mean(x, id_k)) %% # grand mean
mutate(x_w = x - x_c) %% # centering within cluster
mutate(x_m = x - x_g) %% # centering to the grand mean
mutate(x_b = x_c - x_g) %% # centered cluster means
```

```
# [...]
#-----#
# base de datos con diseño
#-----
erce_brr <- survey::svrepdesign(
  data    = data_a6,
  type    = ' Fay',
  rho     = .5,
  weights = ~WT,
  repweights = 'BRR[0-9]+',
  combined.weights = TRUE
)

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

#-
# ajustar modelo
#-
svy_estimates <- survey::svyglm(REPC ~ 1 + x_m + z_w + z_b,
  design = erce_brr,
  family = 'quasibinomial'(link = "logit")
)

#-----#
# extraer estimados a una tabla
#-
tabla_10 <- broom::tidy(svy_estimates, conf.int = TRUE) %>%
  mutate(or = exp(estimate)) %>%
  mutate(or_lo = exp(conf.low)) %>%
  mutate(or_hi = exp(conf.high))

#-
# mostrar tabla
#-
knitr::kable(tabla_10, digits = 2)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high	or	or_lo	or_hi
(Intercept)	-3.94	0.14	-27.51	0.00	-4.23	-3.66	0.02	0.01	0.03
x_m	0.53	0.09	5.66	0.00	0.34	0.72	1.70	1.41	2.05
z_w	-0.68	0.15	-4.48	0.00	-0.98	-0.38	0.51	0.38	0.69
z_b	0.33	0.30	1.10	0.27	-0.26	0.92	1.39	0.77	2.50

Taller

Resumen de modelos ajustados

Códigos empleados para ajustar modelos de regresión lineal y logística

Regresiones Lineales

Las regresiones lineales ajustadas con TSL y valores plausibles, pueden escribirse directamente al interior de la función `mitools:::withPV`.

```
#-----  
# regresion lineal con valores plausibles (TSL)  
  
reg_model_pv <- mitools:::withPV(  
  mapping = mat ~ MAT_1 + MAT_2 + MAT_3 + MAT_4 + MAT_5,  
  data = erce tsl,  
  action = quote(survey::svyglm(mat ~ 1 + ISECF, design = erce_tsl)),  
  rewrite=TRUE  
)
```

En contraste, las regresiones lineales ajustadas con BRR, sobre valores plausibles, requieren que se cree un listado de datos imputados con `mitools:::imputationList`, antes de que se genere el objeto con diseño.

```
#-----  
# regresion lineal con valores plausibles (BRR)  
  
reg_model_pv <- with(erce_brr,  
  survey::svyglm(matpv ~ 1 + ISECF , design = erce_brr  
)
```

Cuando ajustamos un modelo de regresión lineal con TSL, sobre una variable observada basta con preparar los datos. En particular las variables de **clustering**.

```
#-----  
# regresion lineal sobre una variable continua observada (TSL)  
  
reg_model_est <- survey::svyglm(AURES ~ 1 + x_m + z_w + z_b, design = erce_tsl)
```

Cuando ajustamos un modelo de regresión lineal con BRR, sobre una variable observada basta con preparar los datos. En particular las variables de **clustering**; pero además, se requiere que preparemos las variables BRR en caso de que se quiera incluir a más de un país de forma simultánea.

```
#-----  
# regresion lineal sobre una variable continua observada (BRR)  
  
reg_model_est <- survey::svyglm(AURES ~ 1 + x_m + z_w + z_b, design = erce_brr)
```

Regresiones Logísticas

Las regresiones logísticas ajustadas con TSL y valores plausibles, pueden escribirse directamente al interior de la función `mitools:::withPV`.

```
#-----  
# regresion logistica con valores plausibles (TSL)  
  
logit_model_pv <- mitools:::withPV(  
  mapping = mat_mpl ~ mat_mpl1 + mat_mpl2 + mat_mpl3 + mat_mpl4 + mat_mpl5,  
  data = erce_tsl,  
  action = quote(  
    survey::svyglm(mat_mpl ~ 1 + EDU,  
    design = erce_tsl,  
    family = 'quasibinomial'(link = "logit")  
  )),  
  rewrite=TRUE  
)
```

En contraste, las regresiones logísticas ajustadas con BRR, sobre valores plausibles, requieren que se cree un listado de datos imputados con `mitools:::imputationList`, antes de que se genere el objeto con diseño.

```
#-----  
# regresion logistica con valores plausibles (BRR)  
  
logit_model_pv <- with(erce_brr,  
  survey::svyglm(mat_mpl ~ 1 + EDU,  
  design = erce_brr,  
  family = 'quasibinomial'(link = "logit")  
)
```

Cuando ajustamos un modelo de regresión logística con TSL, sobre una variable observada basta con preparar los datos. En particular las variables de **clustering**.

```
#-----  
# regresion logistica variable dicotomico (TSL)  
  
logit_model_est <- survey::svyglm(REPC ~ 1 + x_m + z_w + z_b,  
design = erce_tsl,  
family = 'quasibinomial'(link = "logit")  
)
```

En el caso de una logística con BRR, si ajustamos los datos sobre un solo país, podemos emplear las variables de diseño incluidas en la base de datos. Si es sobre varios países, se requiere preparar las variables de diseño (e.g., clustering y BRR).

```
#-----  
# regresion logistica variable dicotomico (BRR)  
  
logit_model_est <- survey::svyglm(REPC ~ 1 + x_m + z_w + z_b,  
design = erce_brr,  
family = 'quasibinomial'(link = "logit")  
)
```

Muchas gracias!

Referencias

Lumley, T. (2010). Complex Surveys: A Guide to Analysis Using R. Wiley.

Rights, J. D., Preacher, K. J., & Cole, D. A. (2019). The danger of conflating level-specific effects of control variables when primary interest lies in level-2 effects. *British Journal of Mathematical and Statistical Psychology*, 4, bmsp.12194.
<https://doi.org/10.1111/bmsp.12194>