Análisis de encuestas de hogares con R

Modulo 11: Estimación con pesos de replica

CEPAL - Unidad de Estadísticas Sociales

Definición las replica.

Para mejorar la estimación es posible utilizar técnicas de remuestreo, tal es el caso de JKn y bootstrap que ya están incluidas den el paquete de *survey*, para poder ejecutarlas es relativamente simple.

```
encuesta <- readRDS("../Data/encuesta.rds")
diseno <- encuesta %>%
  as_survey_design(
    strata = Stratum,
    ids = PSU,
    weights = wk,
    nest = T
  )
```

Tipos de replicas.

Las funciones que permite incorporar los pesos de replica son `as_survey_rep` y `as.svrepdesign`, las opciones disponibles en este paquete son:

- JK1, JKn Jackknife Repeated Replication (JRR) (Rust, 1985; Wolter, 2007)

Tipos de replicas.

Las funciones que permite incorporar los pesos de replica son `as_survey_rep` y `as.svrepdesign`, las opciones disponibles en este paquete son:

- JK1, JKn Jackknife Repeated Replication (JRR) (Rust, 1985; Wolter, 2007)
- BRR: Balanced Repeated Replication

Tipos de replicas.

Las funciones que permite incorporar los pesos de replica son `as_survey_rep` y `as.svrepdesign`, las opciones disponibles en este paquete son:

- JK1, JKn Jackknife Repeated Replication (JRR) (Rust, 1985; Wolter, 2007)
- BRR: Balanced Repeated Replication
- bootstrap, **subbootstrap**, mrbbootstrap

Tipos de replicas.

Las funciones que permite incorporar los pesos de replica son `as_survey_rep` y `as.svrepdesign`, las opciones disponibles en este paquete son:

- JK1, JKn Jackknife Repeated Replication (JRR) (Rust, 1985; Wolter, 2007)
- BRR: Balanced Repeated Replication
- bootstrap, **subbootstrap**, mrbbootstrap
- Fay

Definiendo el objeto diseño con replicas **Jackknife** , **BRR** y **Fay**


```
set.seed(123)
diseno_JKn <-
  as_survey_rep(diseno,
                type = "JKn"
                )
set.seed(123)
diseno_BRR <-
  as_survey_rep(diseno,
                type = "BRR")
diseno_Fay <-
  as_survey_rep(diseno,
                type="Fay", rho=0.3)
```

Definiendo el objeto diseño con replicas **bootstrap**


```
set.seed(123)
diseno_bootstrap <-
  as_survey_rep(diseno,
                type = "bootstrap",
                replicates = 100 )
set.seed(123)
diseno_subbootstrap <-
  as_survey_rep(diseno,
                type = "subbootstrap",
                replicates = 100)
set.seed(123)
diseno_mrbbootstrap <-
  as_survey_rep(diseno,
                type = "mrbbootstrap",
                replicates = 100)
```

La matriz de replica.

Las definiciones anteriores crean una nueva entrada en el objeto diseño el cual contiene la matriz de replicas (`repweights`). Las columnas de la matriz son el número de replicas, que puede variar según el método, las filas son el número de registros en la encuesta; y el valor de la celda estable por cuanto se debe multiplicar el factor de expansión para realizar la estimación del parámetro. Los valores de la celda pueden ser enteros para indicar ausencia (0) o presencia (1) o repetir el registro (>1), como sucede con BRR, Jackknife y bootstrap. En el caso de Fay, esto no ocurre dado que se trata de fracciones de muestreo que están asociado con el parámetro de ρ.

Comparando estimaciones de la varianza.

La sintaxis muestra que sin importar la metodología de replica el procedimiento es igual.

```
library(purrr)
library(magrittr)
list(
  sin_rep = diseno,
  JKn = diseno_JKn,
  BRR = diseno_BRR,
  Fay = diseno_Fay,
  bootstrap = diseno_bootstrap,
  subbootstrap = diseno_subbootstrap,
  mrbbootstrap = diseno_mrbbootstrap
) %>%
  map(
    ~ .x %>% summarise(
      "Nacional" = survey_mean(Income,deff = TRUE))) %>%
  bind_rows(.id = "Diseno")
```

Comparando estimaciones de la varianza.

La tabla muestra el ingreso medio nacional.

Diseno	Nacional	Nacional_se	Nacional_deff
sin_rep	570.9	28.48	8.821
JKn	570.9	28.48	8.822
BRR	570.9	28.50	8.837
Fay	570.9	28.49	8.827
bootstrap	570.9	28.39	8.766
subbootstrap	570.9	27.41	8.171
mrbbootstrap	570.9	28.24	8.677

Comparando estimaciones de la varianza.

La tabla muestra el ingreso mediano nacional.

Diseno	Nacional	Nacional_se
sin_rep	437.4	31.93
JKn	437.4	31.93
BRR	437.4	31.93
Fay	437.4	31.93
bootstrap	437.4	39.99
subbootstrap	437.4	31.86
mrbbootstrap	437.4	28.66

Comparando estimaciones de la varianza.

La tabla muestra el quantile 25 del ingreso.

Diseno	Nacional_q25	Nacional_q25_se
sin_rep	250	18.92
JKn	250	18.92
BRR	250	18.92
Fay	250	18.92
bootstrap	250	19.00
subbootstrap	250	19.00
mrbbootstrap	250	19.00

Con los resultados mostrados previamente y con la intención de observar diferentes resultados nos decantamos por el **diseno_subbootstrap** para replicar algunos de los análisis realizados en secciones anteriores.

Análisis de variables continuas

Estimación de la media del gasto

Un resultado más interesante para la variable gasto es el promedio de la variable.

```
svymean (~Expenditure,
         diseno_subbootstrap, deff=T) %>%
  data.frame()
```

	mean	SE	deff
Expenditure	370.5	12.07	4.958

```
svymean (~Expenditure,
         diseno, deff=T) %>% data.frame()
```

	mean	Expenditure	deff
Expenditure	370.5	13.29	6.016

Estimación de la media por sub-grupos


```
diseno_subbootstrap %>% group_by(Zone, Sex) %>%
  summarise(
    Media = survey_mean(
      Expenditure, level = 0.95,
       vartype =  c("se"), deff = TRUE)) %>%
  as.data.frame()
```

Estimación de la media por sub-grupos

Resultado con replicas

Zone	Sex	Media	Media_se	Media_deff
Rural	Female	272.7	10.68	2.997
Rural	Male	275.3	10.80	2.510
Urban	Female	450.8	18.39	2.494
Urban	Male	469.8	23.69	2.884

Resultados sin replicas

Zone	Sex	Media	Media_se	Media_deff
Rural	Female	272.7	11.61	3.545
Rural	Male	275.3	10.25	2.260
Urban	Female	450.8	20.12	2.985
Urban	Male	469.8	26.96	3.737

Estimación de la razón entre hombres y mujeres

La estimación de una razón se obtiene con la función `survey_ratio`.

```
diseno_subbootstrap %>% summarise(
    Razon =  survey_ratio(
      numerator = (Sex == "Female"),# creando dummy.
      denominator = (Sex == "Male"),# creando dummy.
      level = 0.95,
    vartype =  c("se", "ci")
    ))
```

Estimación de la razón entre hombres y mujeres

Resultado con replicas

Razon	Razon_se	Razon_low	Razon_upp
1.114	0.0303	1.054	1.174

Resultados sin replicas

Razon	Razon_se	Razon_low	Razon_upp
1.114	0.0351	1.045	1.184

Los resultados muestran que la estimación puntual no cambia, pero vemos una mejora permanente en la estimación de la varianza. # Análisis de variables categóricas

Creación de nuevas variables

Durante los análisis de encuesta surge la necesidad de crear nuevas variables a partir de las existentes, aquí mostramos la definición de algunas de ellas.

```
diseno_subbootstrap <- diseno_subbootstrap %>%
  mutate(
  pobreza = ifelse(Poverty != "NotPoor", 1, 0),
  desempleo =
    ifelse(Employment == "Unemployed", 1, 0))
```

Estimación de proporción de urbano y rural

La función idónea para realizar la estimación de las proporciones es `survey_prop` y la sintaxis es como sigue:

```
(prop_zona2 <- diseno_subbootstrap %>%
   group_by(Zone) %>%
   summarise(
     prop = survey_prop(vartype = c("se"),
                        deff = TRUE )))
```

Zone	prop	prop_se	prop_deff
Rural	0.4798	0.0126	1.677
Urban	0.5202	0.0126	1.677

Proporción de mujeres en la zona urbana y rural


```
(prop_ZonaM_Ocupacion <- diseno_subbootstrap %>%
   filter(Sex == "Female") %>%
   group_by(Zone, Employment) %>%
   summarise(
     prop = survey_prop(
       vartype = c("se"),deff = TRUE)) %>%
   data.frame())
```

Proporción de mujeres en la zona urbana y rural

Zone	Employment	prop	prop_se	prop_deff
Rural	Unemployed	0.0102	0.0062	2.6212
Rural	Inactive	0.4472	0.0351	3.4490
Rural	Employed	0.2400	0.0374	5.2935
Rural	NA	0.3026	0.0309	3.1294
Urban	Unemployed	0.0211	0.0060	1.2330
Urban	Inactive	0.3645	0.0212	1.3626
Urban	Employed	0.3846	0.0162	0.7834
Urban	NA	0.2299	0.0138	0.7537

Tabla cruzada de Zona Vs Sexo

Haciendo uso de la función `group_by` organizada en forma de `data.frame`.

```
(
  prop_sexo_zona <- diseno_subbootstrap %>%
    group_by(pobreza,Sex) %>%
    summarise(
      prop = survey_prop(vartype = c("se"),
                         deff = TRUE)) %>%
    data.frame()
)
```

Tabla cruzada de Zona Vs Sexo

pobreza	Sex	prop	prop_se	prop_deff
0	Female	0.5292	0.0126	1.069
0	Male	0.4708	0.0126	1.069
1	Female	0.5236	0.0168	1.090
1	Male	0.4764	0.0168	1.090

Tablas de doble entrada.

Una alternativa es utilizar la función `svyby` con la siguiente sintaxis.

```
(tab_Sex_Pobr <- svyby(~Sex,
                       ~pobreza,
                       diseno_subbootstrap,
                       svymean,deff = TRUE))
```

	pobreza	SexFemale	SexMale	se1	se2	DEff.SexFemale	DEff.SexMale
0	0	0.5292	0.4708	0.0126	0.0126	1.069	1.069
1	1	0.5236	0.4764	0.0168	0.0168	1.090	1.090

Modelamiento de variables

Modelo lineal de regresión.

Ahora, emplee la función `svyglm` de `survey`

```
fit_svy <- svyglm(Income ~ Expenditure,
                  design = diseno_subbootstrap)
modNul <- svyglm(Income ~ 1,
                 design = diseno_subbootstrap)
s1 <- summary(fit_svy)
s0 <-summary(modNul)
```

Resumen del Modelo

Table 14: Modelo encuesta ponderada, svyglm

	Income
Expenditure	1.263***
	(0.201)
Constant	103.100
	(69.060)
N	2605
AIC	38281.000

***p < .01; **p < .05; *p < .1

Calculo del R^2


```
(R2 = 1-s1$dispersion/s0$dispersion)
```

```
## [1] 0.5094
```

```
n = sum(diseno_subbootstrap$variables$wk)
(R2Adj = 1-((1-R2)*(n-1)/(n-1-1)))
```

```
## [1] 0.5094
```

Modelo de regresión logistica


```
mod_loglin <- svyglm(
  pobreza ~ Sex + Zone + Region,
  family=quasibinomial, design=diseno_subbootstrap)
tidy(mod_loglin)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-0.4082	0.2851	-1.4317	0.1556
SexMale	0.0086	0.1015	0.0852	0.9323
ZoneUrban	-0.4378	0.2221	-1.9711	0.0517
RegionSur	0.0063	0.3318	0.0190	0.9848
RegionCentro	0.1915	0.4559	0.4201	0.6754
RegionOccidente	0.2319	0.3081	0.7528	0.4535
RegionOriente	0.3699	0.4305	0.8592	0.3924

Modelo log lineal ajustado

Intervalos de confianza para los coeficientes del modelo.

```
bind_cols(
  data.frame(exp_estimado = exp(coef(mod_loglin))),
  as.data.frame(exp(confint(mod_loglin)))
)
```

	exp_estimado	2.5 %	97.5 %
(Intercept)	0.6648	0.3774	1.171
SexMale	1.0087	0.8246	1.234
ZoneUrban	0.6454	0.4152	1.003
RegionSur	1.0063	0.5207	1.945
RegionCentro	1.2111	0.4898	2.995
RegionOccidente	1.2611	0.6839	2.325
RegionOriente	1.4476	0.6157	3.404

Modelo gamma ingreso


```
modelo_gamma <- svyglm(formula = Income ~ Age + Sex +
                         Region + Zone,
                         design = diseno_subbootstrap,
                       family = Gamma(link = "inverse"))
broom::tidy(modelo_gamma)
```

Modelo gamma

Estimación de los parámetro del modelo.

term	estimate	std.error	statistic	p.value
(Intercept)	0.0025	2e-04	10.0256	0.0000
Age	0.0000	0e+00	-1.2541	0.2130
SexMale	-0.0001	1e-04	-1.6470	0.1030
RegionSur	0.0000	2e-04	-0.2325	0.8167
RegionCentro	0.0000	2e-04	0.2138	0.8312
RegionOccidente	0.0002	2e-04	0.9896	0.3249
RegionOriente	0.0000	3e-04	-0.0081	0.9936
ZoneUrban	-0.0010	2e-04	-4.3810	0.0000

Modelo multinomial


```
diseno_subbootstrap %>% filter(Age >= 15)%>%
  group_by(Employment) %>%
  summarise(Prop = survey_mean(vartype = c("se")))
```

Employment	Prop	Prop_se
Unemployed	0.0429	0.0079
Inactive	0.3840	0.0147
Employed	0.5731	0.0127

Modelo multinomial


```
library(svyVGAM)
diseno_15 <- diseno_subbootstrap %>%
  filter(Age >= 15)
model_mul <- svy_vglm(
    formula = Employment ~ Age + Sex +
      Region + Zone, design = diseno_15,
     crit = "coef",
    family = multinomial(refLevel = "Unemployed"))
```

Modelo multinomial


```
tab_model <- tidy.svyVGAM(
  x = model_mul,
  exponentiate = FALSE,
  conf.int = FALSE) %>% data.frame()
tab_model
```

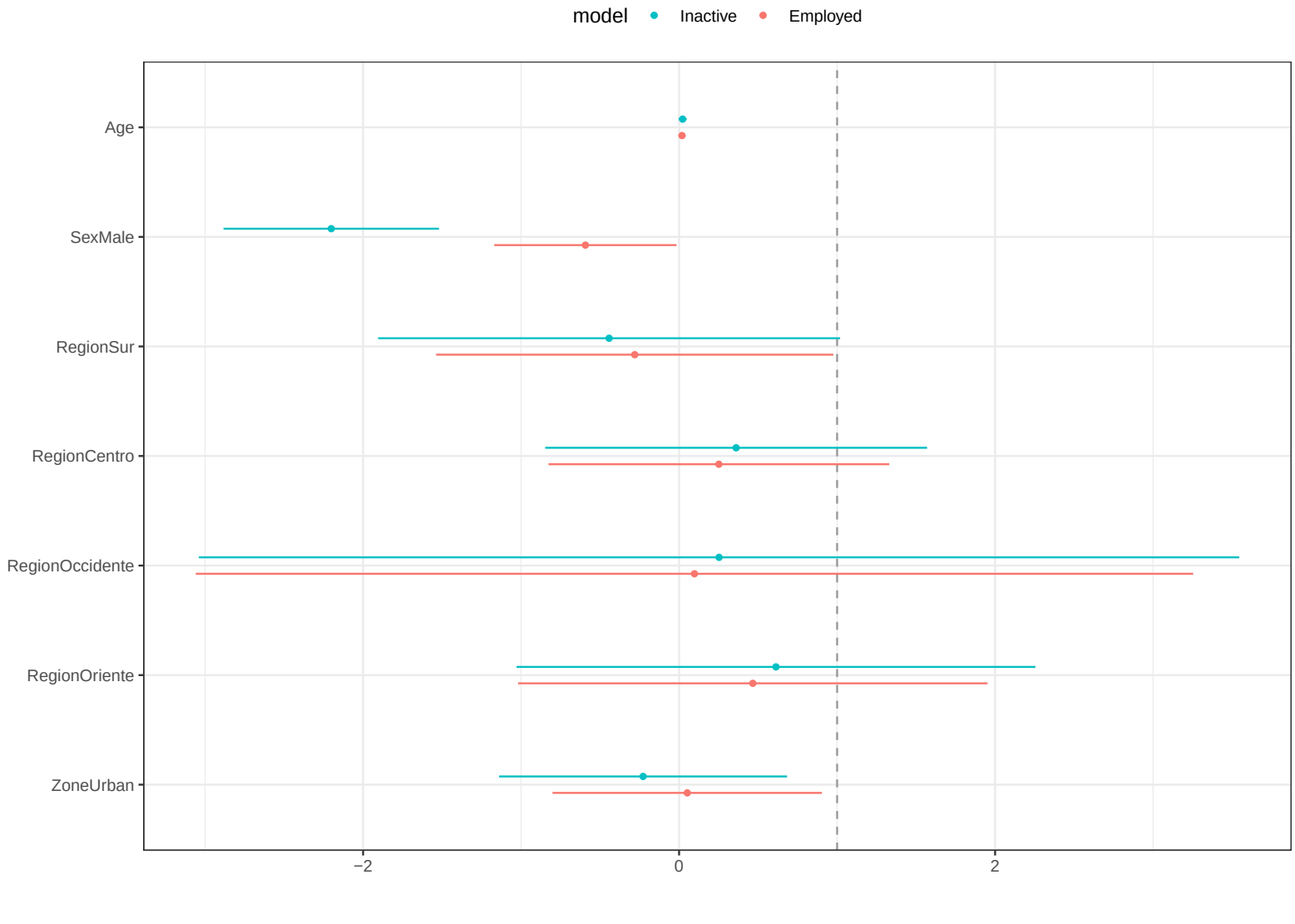
Modelo multinomial

y.level	term	estimate	std.error	statistic	p.value
1	(Intercept)	2.3807	0.9172	2.5956	0.0094
1	Age	0.0222	0.0130	1.7106	0.0872
1	SexMale	-2.2016	0.3480	-6.3271	0.0000
1	RegionSur	-0.4429	0.7460	-0.5938	0.5527
1	RegionCentro	0.3610	0.6167	0.5854	0.5582
1	RegionOccidente	0.2530	1.6797	0.1506	0.8803
1	RegionOriente	0.6130	0.8377	0.7317	0.4643
1	ZoneUrban	-0.2279	0.4650	-0.4900	0.6241
2	(Intercept)	2.1951	0.7707	2.8482	0.0044
2	Age	0.0185	0.0111	1.6685	0.0952
2	SexMale	-0.5930	0.2944	-2.0144	0.0440
2	RegionSur	-0.2809	0.6414	-0.4379	0.6615
2	RegionCentro	0.2517	0.5502	0.4575	0.6473
2	RegionOccidente	0.0973	1.6105	0.0604	0.9518
2	RegionOriente	0.4667	0.7579	0.6158	0.5380
2	ZoneUrban	0.0515	0.4349	0.1185	0.9057

Plot del IC para los coeficientes.


```
tab_model %>%
  mutate(
    model = if_else(
      y.level == 1,
      "Inactive",
      "Employed",
    ),
    sig = gtools::stars.pval(p.value)
  )  %>%
  dotwhisker::dwplot(
    dodge_size = 0.3,
    vline = geom_vline(xintercept = 1, colour = "grey60",
                       linetype = 2)) +
  guides(color = guide_legend(reverse = TRUE)) +
  theme_bw() + theme(legend.position = "top")
```

Plot del IC para los coeficientes.



¡Gracias!

Email: andres.gutierrez@cepal.org