Análisis de encuestas de hogares con R

Módulo 2: Análisis de variables categóricas

CEPAL - Unidad de Estadísticas Sociales

Tabla de contenidos I

Introducción

Motivación

- En el mundo de la estadística y el análisis de datos, nos encontramos con una variedad de variables que pueden ser clasificadas en dos categorías principales: cualitativas y cuantitativas.
- Las variables cualitativas, también conocidas como categóricas, representan características o cualidades que no se pueden medir con números, como el género, el estado civil o el tipo de vivienda.
- Algunas variables cuantitativas se transforman en categóricas al dividir su rango en categorías, y viceversa, algunas variables categóricas se convierten en cuantitativas mediante análisis especializados.
- En esta presentación, exploraremos esta distinción y cómo abordar variables cualitativas en el contexto de encuestas y análisis de datos.

Definición del diseño y creación de variables categóricas

Lectura de la base

Iniciemos con la lectura de la encuesta.

```
encuesta <- readRDS("../Data/encuesta.rds")
```

El paso siguiente es realizar declaración del objeto tipo diseño.

```
options(survey.lonely.psu = "adjust")
library(srvyr)
diseno <- encuesta %>%    # Base de datos.
  as_survey_design(
    strata = Stratum,     # Id de los estratos.
    ids = PSU,            # Id para las observaciones.
    weights = wk,         # Factores de expansión.
    nest = TRUE           # Valida el anidado dentro del estrato
  )
```

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 <- diseno %>% mutate(
  pobreza = ifelse(Poverty != "NotPoor", 1, 0),
  desempleo = ifelse(Employment == "Unemployed", 1, 0),
  edad_18 = case_when(Age < 18 ~ "< 18 anios",
                      TRUE ~ ">= 18 anios")
)
```

Se ha introducido la función `case_when` la cual es una extensión del a función `ifelse` que permite crear múltiples categorías a partir de una o varias condiciones.

Dividiendo la muestra en Sub-grupos

En ocasiones se desea realizar estimaciones por sub-grupos de la población, en este caso se extraer 4 sub-grupos de la encuesta.

```
sub_Urbano <- diseno %>%  filter(Zone == "Urban")
sub_Rural  <- diseno %>%  filter(Zone == "Rural")
sub_Mujer  <- diseno %>%  filter(Sex == "Female")
sub_Hombre <- diseno %>%  filter(Sex == "Male")
```

Estimación del tamaño.

El primer parámetro estimado serán los tamaños de la población y subpoblaciones.

```
(tamano_zona <- diseno %>% group_by(Zone) %>%
   summarise(
     n = unweighted(n()), # Observaciones en la muestra.
     Nd = survey_total(vartype = c("se","ci"))))
```

Zone	n	Nd	Nd_se	Nd_low	Nd_upp
Rural	1297	72102	3062	66039	78165
Urban	1308	78164	2847	72526	83802

En la tabla *n* denota el número de observaciones en la muestra por Zona y *Nd* denota la estimación del total de observaciones en la población.

Estimación de tamaño

Empleando una sintaxis similar es posible estimar el número de personas en condición de pobreza extrema, pobreza y no pobres.

```
(tamano_pobreza <- diseno %>% group_by(Poverty) %>%
   summarise(
       Nd = survey_total(vartype = c("se","ci"))))
```

Poverty	Nd	Nd_se	Nd_low	Nd_upp
NotPoor	91398	4395	82696	100101
Extreme	21519	4949	11719	31319
Relative	37349	3695	30032	44666

Estimación de tamaño

En forma similar es posible estimar el número de personas debajo de la linea de pobreza.

```
(tamano_pobreza <- diseno %>%
   group_by(pobreza) %>%
   summarise(
       Nd = survey_total(vartype = c("se","ci"))))
```

pobreza	Nd	Nd_se	Nd_low	Nd_upp
0	91398	4395	82696	100101
1	58868	5731	47519	70216

Estimación de tamaño

Otra variable de interés es conocer el estado de ocupación de la personas.

```
(tamano_ocupacion <- diseno %>%
   group_by(Employment) %>%
   summarise(
       Nd = survey_total(vartype = c("se","ci"))))
```

Employment	Nd	Nd_se	Nd_low	Nd_upp
Unemployed	4635	760.6	3129	6141
Inactive	41465	2162.8	37183	45748
Employed	61877	2540.1	56847	66907
NA	42289	2780.0	36784	47794

Estimación de tamaño

Utilizando la función `group_by` es posible obtener resultados por más de un nivel de agregación.

```
(tamano_ocupacion_pobreza <- diseno %>%
   group_by(Employment, Poverty) %>%
   cascade(
       Nd = survey_total(vartype = c("se","ci")),
       .fill = "Total") %>%
   data.frame()
   )
```

Estimación de tamaño

Employment	Poverty	Nd	Nd_se	Nd_low	Nd_upp
Unemployed	NotPoor	1768	405.4	965.7	2571
Unemployed	Extreme	1169	348.1	479.9	1859
Unemployed	Relative	1697	457.8	790.7	2604
Unemployed	Total	4635	760.6	3128.7	6141
Inactive	NotPoor	24346	1736.3	20908.0	27784
Inactive	Extreme	6422	1320.7	3806.6	9037
Inactive	Relative	10697	1460.3	7805.9	13589
Inactive	Total	41465	2162.8	37182.7	45748
Employed	NotPoor	44600	2596.2	39459.6	49741
Employed	Extreme	5128	1121.6	2906.6	7349
Employed	Relative	12149	1346.6	9482.7	14816
Employed	Total	61877	2540.1	56847.4	66907
Total	Total	150266	4181.4	141986.5	158546
NA	NotPoor	20684	1256.6	18195.4	23172
NA	Extreme	8800	2979.9	2899.7	14701
NA	Relative	12805	1551.0	9733.9	15876
NA	Total	42289	2780.0	36784.3	47794

Estimación de Proporciones Poblacionales

En encuestas de hogares, a menudo es importante estimar la proporción de una característica particular en una población, como la proporción de personas que tienen un cierto nivel de educación, la proporción de hogares con acceso a servicios básicos, entre otros.

La estimación de una proporción poblacional se puede hacer utilizando la siguiente ecuación:

$$\hat{\pi} = p = \frac{\sum_{i=1}^{n} \omega_i y_i}{\sum_{i=1}^{n} \omega_i}$$

Donde:

- $\hat{\pi}$ es la estimación de la proporción poblacional.
- n es el tamaño de la muestra.
- ω_i son los pesos de muestreo para cada unidad de la muestra.
- y_i es la variable binaria que indica si la unidad de muestreo tiene la característica de interés (1 si la tiene, 0 si no la tiene).

Estimación de proporción de urbano y rural

El procedimiento estándar para el calculo de proporciones es crear una *variable dummy* y sobre está realizar las operaciones. Sin embargo, la librería `srvy` nos simplifica el calculo, mediante la sintaxis.

```
(prop_zona <- diseno %>% group_by(Zone) %>%
   summarise(
     prop = survey_mean(vartype = c("se","ci"),
                        proportion = TRUE )))
```

Zone	prop	prop_se	prop_low	prop_upp
Rural	0.4798	0.014	0.4523	0.5075
Urban	0.5202	0.014	0.4925	0.5477

Note que, se utilizo la función `survey_mean` para la estimación.

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 %>% group_by(Zone) %>%
   summarise(
     prop = survey_prop(vartype = c("se","ci") )))
```

Zone	prop	prop_se	prop_low	prop_upp
Rural	0.4798	0.014	0.4523	0.5075
Urban	0.5202	0.014	0.4925	0.5477

Proporción de hombres y mujeres en la zona urbana

Si el interés es obtener la estimación para una subpoblación, procedemos así:

```
(prop_sexoU <- sub_Urbano %>% group_by(Sex) %>%
   summarise(
       prop = survey_prop(vartype = c("se","ci"))))
```

Sex	prop	prop_se	prop_low	prop_upp
Female	0.5367	0.013	0.5107	0.5625
Male	0.4633	0.013	0.4375	0.4893

¿Cómo estimar el Proporción de hombres dado que están en zona rural?

Proporción de hombres y mujeres en la zona rural

```
(prop_sexoR <- sub_Rural %>% group_by(Sex) %>%
   summarise(
     n = unweighted(n()),
     prop = survey_prop(vartype = c("se","ci"))))
```

Sex	n	prop	prop_se	prop_low	prop_upp
Female	679	0.5165	0.0082	0.4999	0.5330
Male	618	0.4835	0.0082	0.4670	0.5001

¿Cómo estimar el Proporción de hombres en la zona rural dado que es hombre?

Proporción de hombres en la zona urbana y rural

```
(prop_ZonaH <- sub_Hombre %>% group_by(Zone) %>%
   summarise(
     prop = survey_prop(vartype = c("se","ci"))))
```

Zone	prop	prop_se	prop_low	prop_upp
Rural	0.4905	0.0178	0.4553	0.5258
Urban	0.5095	0.0178	0.4742	0.5447

¿Cómo estimar el Proporción de mujeres en la zona rural dado que es mujer?

Proporción de mujeres en la zona urbana y rural

```
(prop_ZonaM <- sub_Mujer %>% group_by(Zone) %>%
   summarise(
    prop = survey_prop(vartype = c("se","ci"))))
```

Zone	prop	prop_se	prop_low	prop_upp
Rural	0.4702	0.014	0.4426	0.4981
Urban	0.5298	0.014	0.5019	0.5574

Proporción de hombres en la zona urbana y rural

Con el uso de la función `group_by` es posible estimar un mayor numero de niveles de agregación al combinar dos o más variables.

```
(prop_ZonaH_Pobreza <- sub_Hombre %>%
   group_by(Zone, Poverty) %>%
   summarise(
     prop = survey_prop(vartype = c("se","ci")))%>%
   data.frame())
```

Proporción de hombres en la zona urbana y rural

Zone	Poverty	prop	prop_se	prop_low	prop_upp
Rural	NotPoor	0.5488	0.0626	0.4243	0.6675
Rural	Extreme	0.1975	0.0675	0.0958	0.3637
Rural	Relative	0.2536	0.0372	0.1871	0.3341
Urban	NotPoor	0.6599	0.0366	0.5842	0.7283
Urban	Extreme	0.1129	0.0245	0.0726	0.1712
Urban	Relative	0.2272	0.0260	0.1798	0.2828

Proporción de mujeres en la zona urbana y rural

```
(prop_ZonaM_Pobreza <- sub_Mujer %>%
   group_by(Zone, Poverty) %>%
   summarise(
     prop = survey_prop(vartype = c("se","ci"))) %>%
   data.frame())
```

Zone	Poverty	prop	prop_se	prop_low	prop_upp
Rural	NotPoor	0.5539	0.0557	0.4428	0.6599
Rural	Extreme	0.1600	0.0557	0.0773	0.3022
Rural	Relative	0.2861	0.0436	0.2080	0.3794
Urban	NotPoor	0.6612	0.0322	0.5948	0.7219
Urban	Extreme	0.1094	0.0221	0.0727	0.1614
Urban	Relative	0.2294	0.0266	0.1811	0.2861

Proporción de hombres en la zona y empleado

```
(prop_ZonaH_Ocupacion <- sub_Hombre %>%
   group_by(Zone, Employment) %>%
   summarise(
     prop = survey_prop(vartype = c("se","ci")))%>%
   data.frame())
```

Zone	Employment	prop	prop_se	prop_low	prop_upp
Rural	Unemployed	0.0513	0.0157	0.0277	0.0930
Rural	Inactive	0.1035	0.0203	0.0697	0.1511
Rural	Employed	0.5225	0.0265	0.4699	0.5746
Rural	NA	0.3227	0.0350	0.2576	0.3955
Urban	Unemployed	0.0437	0.0085	0.0297	0.0640
Urban	Inactive	0.1633	0.0181	0.1306	0.2024
Urban	Employed	0.5134	0.0236	0.4666	0.5599
Urban	NA	0.2796	0.0221	0.2380	0.3253

Proporción de mujeres en la zona urbana y rural

```
(prop_ZonaM_Ocupacion <- sub_Mujer %>% group_by(Zone, Employment) %>%
   summarise( prop = survey_prop(vartype = c("se","ci"))) %>%
   data.frame())
```

Zone	Employment	prop	prop_se	prop_low	prop_upp
Rural	Unemployed	0.0102	0.0055	0.0034	0.0296
Rural	Inactive	0.4472	0.0352	0.3789	0.5176
Rural	Employed	0.2400	0.0392	0.1711	0.3257
Rural	NA	0.3026	0.0308	0.2454	0.3668
Urban	Unemployed	0.0211	0.0060	0.0120	0.0368
Urban	Inactive	0.3645	0.0214	0.3231	0.4079
Urban	Employed	0.3846	0.0195	0.3468	0.4237
Urban	NA	0.2299	0.0139	0.2036	0.2585

Estimación de la proporción de personas menor a 18 años

```
diseno %>%  group_by(edad_18, pobreza) %>%
  summarise(Prop = survey_prop(vartype =  c("se", "ci"))) %>%
  data.frame()
```

edad_18	pobreza	Prop	Prop_se	Prop_low	Prop_upp
< 18 anios	0	0.4985	0.0373	0.4252	0.5718
< 18 anios	1	0.5015	0.0373	0.4282	0.5748
>= 18 anios	0	0.6646	0.0298	0.6033	0.7208
>= 18 anios	1	0.3354	0.0298	0.2792	0.3967

Estimación de la proporción de personas menor a 18 años

```
diseno %>% group_by(edad_18, desempleo) %>%
  summarise( Prop = survey_prop(
      vartype =  c("se", "ci"))) %>% data.frame()
```

edad_18	desempleo	Prop	Prop_se	Prop_low	Prop_upp
< 18 anios	0	0.1667	0.0149	0.1393	0.1982
< 18 anios	1	0.0037	0.0020	0.0013	0.0106
< 18 anios	NA	0.8296	0.0150	0.7978	0.8573
>= 18 anios	0	0.9552	0.0076	0.9377	0.9680
>= 18 anios	1	0.0448	0.0076	0.0320	0.0623

Estimación de la proporción de personas menor a 18 años en zona rural

```
sub_Rural %>%
  group_by(edad_18) %>%
  summarise(
    Prop = survey_prop(
      vartype =  c("se", "ci"))) %>%
  data.frame()
```

edad_18	Prop	Prop_se	Prop_low	Prop_upp
< 18 anios	0.3712	0.0302	0.3129	0.4335
>= 18 anios	0.6288	0.0302	0.5665	0.6871

Estimación de la proporción de mujeres rango de edad

```
sub_Mujer %>% mutate(
  edad_rango = case_when(
  Age>= 18 & Age <=35  ~ "18 - 35",
  TRUE ~ "Otro")) %>% group_by(edad_rango, Employment) %>%
  summarise(Prop = survey_prop(
      vartype =  c("se", "ci"))) %>%   data.frame()
```

edad_rango	Employment	Prop	Prop_se	Prop_low	Prop_upp
18 - 35	Unemployed	0.0289	0.0091	0.0154	0.0537
18 - 35	Inactive	0.5165	0.0379	0.4417	0.5907
18 - 35	Employed	0.4545	0.0357	0.3852	0.5256
Otro	Unemployed	0.0102	0.0040	0.0046	0.0222
Otro	Inactive	0.3527	0.0207	0.3128	0.3947
Otro	Employed	0.2548	0.0217	0.2143	0.3001
Otro	NA	0.3823	0.0223	0.3392	0.4273

Estimación de la proporción de hombres rango de edad

```
sub_Hombre %>%  mutate(
  edad_rango = case_when(
  Age>= 18 & Age <=35  ~ "18 - 35",
  TRUE ~ "Otro")) %>% group_by(edad_rango, Employment) %>%
  summarise(Prop = survey_prop(
      vartype =  c("se", "ci"))) %>%   data.frame()
```

edad_rango	Employment	Prop	Prop_se	Prop_low	Prop_upp
18 - 35	Unemployed	0.0964	0.0182	0.0658	0.1390
18 - 35	Inactive	0.0894	0.0164	0.0618	0.1277
18 - 35	Employed	0.8142	0.0230	0.7644	0.8555
Otro	Unemployed	0.0261	0.0072	0.0151	0.0447
Otro	Inactive	0.1534	0.0199	0.1181	0.1971
Otro	Employed	0.3885	0.0203	0.3492	0.4293
Otro	NA	0.4320	0.0211	0.3908	0.4742

Tablas cruzadas.

Tablas Cruzadas en el Análisis de Encuestas de Hogares

- Las tablas cruzadas son una herramienta esencial.
- Se utilizan para resumir información de variables categóricas.
- Pueden tener dos o más filas y columnas.
- En esta sección, nos enfocaremos principalmente en tablas 2×2.

Estructura de una Tabla de Contingencia:

- Se asume como un arreglo bidimensional de filas y columnas.
- Marginales de fila y columna se calculan sumando las frecuencias.

Ejemplo Gráfico de una Tabla 2×2.

Representa la relación entre dos variables categóricas.

Variable 2	Variable 1	Marginal fila	
	0	1	
0	n_{00}	n_{01}	n_{0+}
1	n_{10}	n_{11}	n_{1+}
Marginal columna	n_{+0}	n_{+1}	n_{++}

Las frecuencias en la tabla pueden ser estimadas o ponderadas. Por ejemplo, $\hat{N}_{01}$ se calcula como la suma ponderada.

Cálculo de Proporciones Estimadas

Las proporciones se obtienen dividiendo las frecuencias ponderadas por el total. Por ejemplo:

$$p_{rc} = \frac{\hat{N}_{rc}}{\hat{N}_{++}}$$

.

Estas tablas cruzadas son fundamentales para explorar la relación entre diferentes variables categóricas en encuestas de hogares y extraer información valiosa para la toma de decisiones.

Estimación de Proporciones para Variables Binarias

- La estimación de una sola proporción, π, para una variable binaria se relaciona con el estimador de razón.
- Al recodificar las respuestas en 0 y 1, podemos estimar la proporción π.
- El estimador de proporción es:

$$p = \frac{\sum_{h=1}^{H}\sum_{\alpha=1}^{\alpha_h}\sum_{i\in(0,1)}^{n_{h\alpha}} \omega_{h\alpha i} I\left(y_i = 1\right)}{\sum_{h=1}^{H}\sum_{\alpha=1}^{\alpha_h}\sum_{i\in(0,1)}^{n_{h\alpha}} \omega_{h\alpha i}} = \frac{\hat{N}_1}{\hat{N}}$$

Estimación de la varianza $\hat{v}(p)$

- La varianza del estimador se calcula con Linealización de Taylor:

$$\hat{v}(p) \approx \frac{V(\hat{N}_1) + p^2 V(\hat{N}) - 2p\, cov(\hat{N}_1, \hat{N})}{\hat{N}^2}$$

- Para evitar límites no interpretables en el intervalo de confianza cuando p está cerca de 0 o 1, podemos utilizar el método de *Wilson modificado*.
- El intervalo de confianza se calcula a través de la transformación Logit:

$$IC[logit(p)] = \left\{ ln\left(\frac{p}{1-p}\right) \pm \frac{t_{1-\alpha/2,gl}\, se(p)}{p(1-p)} \right\}$$

Estimación de la varianza $IC(p)$

El intervalo de confianza para p es:

$$IC(p) = \left\{ \frac{exp\left[ln\left(\frac{p}{1-p}\right) \pm \frac{t_{1-\alpha/2,gl}\, se(p)}{p(1-p)}\right]}{1 + exp\left[ln\left(\frac{p}{1-p}\right) \pm \frac{t_{1-\alpha/2,gl}\, se(p)}{p(1-p)}\right]} \right\}$$

Estimación de Proporciones para Variables Multinomiales

- Cuando se trabaja con variables multinomiales, el estimador de proporción se adapta.
- El estimador para la categoría k es:

$$p_k = \frac{\sum_{h=1}^{H}\sum_{\alpha=1}^{\alpha_h}\sum_{i=1}^{n_{h\alpha}}\omega_{h\alpha i} I\left(y_i = k\right)}{\sum_{h=1}^{H}\sum_{\alpha=1}^{\alpha_h}\sum_{i=1}^{n_{h\alpha}}\omega_{h\alpha i}} = \frac{\widehat{N}_k}{\widehat{N}}$$

Estos métodos permiten estimar proporciones para variables binarias y multinomiales en el contexto de encuestas de hogares.

Tabla Zona Vs Sexo

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

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

Esta forma de organizar la información es recomendable cuando el realizar el análisis sobre las estimaciones puntuales.

Tabla Zona Vs Sexo

pobreza	Sex	prop	prop_se	prop_low	prop_upp
0	Female	0.5292	0.0124	0.5045	0.5537
0	Male	0.4708	0.0124	0.4463	0.4955
1	Female	0.5236	0.0159	0.4922	0.5549
1	Male	0.4764	0.0159	0.4451	0.5078

Tablas de doble entrada.

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

```
tab_Sex_Pobr <- svyby(~Sex,  ~pobreza, diseno, svymean)
tab_Sex_Pobr %>% select(-se.SexFemale, -se.SexMale)
```

	pobreza	SexFemale	SexMale
0	0	0.5292	0.4708
1	1	0.5236	0.4764

```
tab_Sex_Pobr %>% select(-SexFemale, -SexMale)
```

	pobreza	se.SexFemale	se.SexMale
0	0	0.0124	0.0124
1	1	0.0159	0.0159

Tablas de doble entrada.

Para la estimación de los intervalos de confianza utilizar la función `confint`.

```
confint(tab_Sex_Pobr) %>% as.data.frame()
```

	2.5 %	97.5 %
0:SexFemale	0.5048	0.5535
1:SexFemale	0.4925	0.5547
0:SexMale	0.4465	0.4952
1:SexMale	0.4453	0.5075

Prueba de independencia χ^2

- La prueba de independencia χ^2 se utiliza para determinar si dos variables cualitativas son independientes o si hay una asociación entre ellas.
- La prueba se aplica comúnmente a tablas de contingencia, especialmente las 2×2.
- La fórmula para el estadístico χ^2 de Pearson es:

$$\chi^2 = n_{++} \sum_r \sum_c \frac{(p_{rc} - \hat{\pi}_{rc})^2}{\hat{\pi}_{rc}}$$

- Donde $\hat{\pi}_{rc}$ se calcula como:

$$\hat{\pi}_{rc} = \frac{n_{r+}}{n_{++}} \cdot \frac{n_{+c}}{n_{++}} \cdot p_{r+} \cdot p_{+c}$$

Prueba de independencia.

Para realizar la prueba de independencia χ^2 puede ejecuta la siguiente linea de código.

```
svychisq(~Sex + pobreza, diseno, statistic="F")
```

```
    Pearson's X^2: Rao & Scott adjustment

data:  NextMethod()
F = 0.056, ndf = 1, ddf = 119, p-value = 0.8
```

Más adelante se profundiza en la metodología de esta prueba.

Tablas de doble entrada.

```
(tab_Sex_Ocupa <- svyby(~Sex,  ~Employment,
                        diseno, svymean))
```

	Employment	SexFemale	SexMale	se.SexFemale	se.SexMale
Unemployed	Unemployed	0.2727	0.7273	0.0535	0.0535
Inactive	Inactive	0.7703	0.2297	0.0234	0.0234
Employed	Employed	0.4052	0.5948	0.0185	0.0185

Tablas de doble entrada

```
confint(tab_Sex_Ocupa) %>% as.data.frame()
```

	2.5 %	97.5 %
Unemployed:SexFemale	0.1678	0.3776
Inactive:SexFemale	0.7245	0.8162
Employed:SexFemale	0.3689	0.4415
Unemployed:SexMale	0.6224	0.8322
Inactive:SexMale	0.1838	0.2755
Employed:SexMale	0.5585	0.6311

Prueba de independencia.

La prueba de independencia χ^2 se obtiene con la siguiente linea de código.

```
svychisq(~Sex + Employment,
         design = diseno,  statistic="F")
```

```
    Pearson's X^2: Rao & Scott adjustment

data:  NextMethod()
F = 62, ndf = 1.7, ddf = 200.7, p-value <2e-16
```

Tablas de doble entrada.

Dado que la variable *pobreza* es de tipo numérica, es necesario convertirla en factor.

```
tab_region_pobreza <-
  svyby(~as.factor(pobreza),  ~Region, diseno, svymean)
tab_region_pobreza %>% select(-"se.as.factor(pobreza)0",
         -"se.as.factor(pobreza)1")
```

	Region	as.factor(pobreza)0	as.factor(pobreza)1
Norte	Norte	0.6410	0.3590
Sur	Sur	0.6562	0.3438
Centro	Centro	0.6346	0.3654
Occidente	Occidente	0.5992	0.4008
Oriente	Oriente	0.5482	0.4518

Tablas de doble entrada.

```
tab_region_pobreza %>%
 select("se.as.factor(pobreza)0",
        "se.as.factor(pobreza)1")
```

	se.as.factor(pobreza)0	se.as.factor(pobreza)1
Norte	0.0555	0.0555
Sur	0.0435	0.0435
Centro	0.0786	0.0786
Occidente	0.0467	0.0467
Oriente	0.0885	0.0885

Prueba de independencia.

Una vez más la prueba de independencia es:

```
svychisq(~Region + pobreza,
         design = diseno,  statistic="F")
```

```
    Pearson's X^2: Rao & Scott adjustment

data:  NextMethod()
F = 0.49, ndf = 3, ddf = 358, p-value = 0.7
```

Razón de odds

- La razón de odds es una medida que expresa la probabilidad de que un evento ocurra en comparación con la probabilidad de que no ocurra.
- Es una forma de cuantificar la asociación entre los niveles de una variable y un factor categórico.
- La razón de odds se calcula como la proporción de la probabilidad de éxito (ocurrencia del evento) sobre la probabilidad de fracaso (no ocurrencia del evento).
- La fórmula general para la razón de odds es:

$$Odds = \frac{P(\text{Éxito})}{P(\text{Fracaso})}$$

- Se utiliza comúnmente en estadística y análisis de datos para evaluar la asociación entre variables y en modelos de regresión logística.

Razón de obbs

```
(tab_Sex <- svyby(~pobreza,  ~Sex, diseno,
                  svymean, vartype = c("se", "ci")))
```

	Sex	pobreza	se	ci_l	ci_u
Female	Female	0.3892	0.0316	0.3273	0.4512
Male	Male	0.3946	0.0366	0.3228	0.4664

```
svycontrast(tab_Sex, quote(`Female`/`Male`)  )
```

```
         nlcon   SE
contrast 0.987 0.12
```

Razón de obbs

```
tab_Sex_Pobr <-
   svymean(~interaction (Sex, pobreza), diseno,
             se=T, na.rm=T, ci=T, keep.vars=T)
  tab_Sex_Pobr %>%  as.data.frame()
```

	mean	SE
interaction(Sex, pobreza)Female.0	0.3219	0.0178
interaction(Sex, pobreza)Male.0	0.2864	0.0177
interaction(Sex, pobreza)Female.1	0.2051	0.0166
interaction(Sex, pobreza)Male.1	0.1866	0.0178

Razón de obbs

Suponga que se desea cálcular la siguiente razón de obbs.

$$\frac{\frac{P(Sex{=}Female|pobreza{=}0)}{P(Sex{=}Female|pobreza{=}1)}}{\frac{P(Sex{=}Male|pobreza{=}1)}{P(Sex{=}Male|pobreza{=}0)}}$$

La forma de cálculo en sería:

```
svycontrast(tab_Sex_Pobr,
            quote((`interaction(Sex, pobreza)Female.0`/
                     `interaction(Sex, pobreza)Female.1`)/
                   (`interaction(Sex, pobreza)Male.0`/
                     `interaction(Sex, pobreza)Male.1`) ))
```

```
         nlcon  SE
contrast  1.02 0.1
```

Diferencia de proporciones en tablas de contingencias

Como lo menciona *Heeringa, S. G. (2017)* las estimaciones de las proporciones de las filas en las tablas de doble entrada son, de hecho, estimaciones de subpoblaciones en las que la subpoblación se define por los niveles de la variable factorial. Ahora bien, si el interés se centra en estimar diferencias de las proporciones de las categorías entre dos niveles de una variable factorial, se pueden utilizando contrastes.

Contrastes

El interés ahora es realizar en contraste de proporciones, por ejemplo: $\hat{p}_F - \hat{p}_M$

```
(tab_sex_pobreza <- svyby(~pobreza, ~Sex,
                          diseno ,
                        svymean, na.rm=T,
                        covmat = TRUE,
                        vartype = c("se", "ci")))
```

	Sex	pobreza	se	ci_l	ci_u
Female	Female	0.3892	0.0316	0.3273	0.4512
Male	Male	0.3946	0.0366	0.3228	0.4664

- *Paso 1:* Calcular la diferencia de estimaciones

```
0.3892 - 0.3946
```

```
[1] -0.0054
```

Contrastes de la diferencia de proporciones

Con la función `vcov` obtiene la matriz de covarianzas

```
library(kableExtra)
vcov(tab_sex_pobreza)%>% data.frame() %>%
  kable(digits = 10,
        format.args = list(scientific = FALSE))
```

	Female	Male
Female	0.0009983	0.0009183
Male	0.0009183	0.0013416

- *Paso 2:* El cálculo del error estándar es:

```
sqrt(0.0009983 + 0.0013416 - 2*0.0009183)
```

```
[1] 0.02243
```

Contrastes de la diferencia de proporciones en R

Para realizar la diferencia de proporciones se hace uso de la función `svycontrast`.

```
svycontrast(tab_sex_pobreza,
            list(diff_Sex = c(1, -1))) %>%
  data.frame()
```

	contrast	diff_Sex
diff_Sex	-0.0053	0.0224

Contrastes de la diferencia de proporciones

Diferencia en desempleo por sexo.

```
(tab_sex_desempleo <- svyby(
  ~desempleo, ~Sex,
    diseno %>% filter(!is.na(desempleo)) ,
     svymean, na.rm=T, covmat = TRUE,
     vartype = c("se", "ci")))
```

	Sex	desempleo	se	ci_l	ci_u
Female	Female	0.0217	0.0056	0.0107	0.0326
Male	Male	0.0678	0.0122	0.0440	0.0917

- *Paso 1*: Diferencia de las estimaciones

```
0.02169 - 0.06783
```

```
[1] -0.04614
```

Contrastes de la diferencia de proporciones

Estimación de la matriz de covarianza:

```
vcov(tab_sex_desempleo) %>% data.frame() %>%
  kable(digits = 10,
        format.args = list(scientific = FALSE))
```

	Female	Male
Female	0.00003114	0.00002081
Male	0.00002081	0.00014789

- *Paso 2*: Estimación del error estándar.

```
sqrt(0.00003114  + 0.00014789 - 2*0.00002081)
```

```
[1] 0.01172
```

Contrastes de la diferencia de proporciones en R

Siguiendo el ejemplo anterior se tiene que:

```
svycontrast(tab_sex_desempleo,
            list(diff_Sex = c(-1, 1))) %>%
  data.frame()
```

	contrast	diff_Sex
diff_Sex	0.0461	0.0117

Contrastes

La tabla de desempleo por región se obtiene como:

```
(tab_region_desempleo <- svyby(
  ~desempleo, ~Region,
    diseno %>% filter(!is.na(desempleo)) ,
    svymean, na.rm=T, covmat = TRUE,
     vartype = c("se", "ci")))
```

	Region	desempleo	se	ci_l	ci_u
Norte	Norte	0.0488	0.0200	0.0095	0.0880
Sur	Sur	0.0656	0.0238	0.0191	0.1122
Centro	Centro	0.0387	0.0124	0.0144	0.0630
Occidente	Occidente	0.0400	0.0123	0.0159	0.0641
Oriente	Oriente	0.0295	0.0126	0.0049	0.0541

Creado una matriz de contrastes

Ahora, el interés es realizar los contrastes siguientes para desempleo:

- $\hat{p}_{Norte} - \hat{p}_{Centro} = 0.01004$,
- $\hat{p}_{Sur} - \hat{p}_{Centro} = 0.02691$
- $\hat{p}_{Occidente} - \hat{p}_{Oriente} = 0.01046$

Escrita de forma matricial es:

$$\begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Contrastes múltiples

```
vcov(tab_region_desempleo)%>%
  data.frame() %>%
  kable(digits = 10,
        format.args = list(scientific = FALSE))
```

	Norte	Sur	Centro	Occidente	Oriente
Norte	0.0004009	0.0000000	0.0000000	0.0000000	0.000000
Sur	0.0000000	0.0005641	0.0000000	0.0000000	0.000000
Centro	0.0000000	0.0000000	0.0001538	0.0000000	0.000000
Occidente	0.0000000	0.0000000	0.0000000	0.0001512	0.000000
Oriente	0.0000000	0.0000000	0.0000000	0.0000000	0.000158

```
c(sqrt(0.0002981 + 0.0002884 - 2*0) ,sqrt(0.0001968 + 0.0002884 - 2*0),
sqrt(0.0001267 + 0.0004093 - 2*0))
```

```
[1] 0.02422 0.02203 0.02315
```

Creado una matriz de contrastes en R

```
svycontrast(tab_region_desempleo, list(
           Norte_sur = c(1, 0, -1, 0, 0),
         Sur_centro = c(0, 1, -1, 0, 0),
   Occidente_Oriente = c(0, 0, 0, 1, -1)
            )) %>% data.frame()
```

	contrast	SE
Norte_sur	0.0100	0.0236
Sur_centro	0.0269	0.0268
Occidente_Oriente	0.0105	0.0176

Ejercicio.

Repetir el contraste anterior para pobreza.

```
(tab_region_pobreza <- svyby(
  ~pobreza, ~Region,
    diseno %>% filter(!is.na(desempleo)) ,
     svymean, na.rm=T, covmat = TRUE,
      vartype = c("se", "ci")))
```

	Region	pobreza	se	ci_l	ci_u
Norte	Norte	0.3263	0.0480	0.2322	0.4204
Sur	Sur	0.2947	0.0479	0.2007	0.3886
Centro	Centro	0.3234	0.0721	0.1820	0.4647
Occidente	Occidente	0.3673	0.0440	0.2811	0.4536
Oriente	Oriente	0.3871	0.0916	0.2075	0.5666

Creado una matriz de contrastes

Ahora, el interés es realizar los contrastes siguientes para pobreza:

- $\hat{p}_{Norte} - \hat{p}_{Centro}$,
- $\hat{p}_{Sur} - \hat{p}_{Centro}$
- $\hat{p}_{Occidente} - \hat{p}_{Oriente}$

Escrita de forma matricial es:

$$\begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Creado una matriz de contrastes en R

```
svycontrast(tab_region_pobreza, list(
           Norte_sur = c(1, 0, -1, 0, 0),
         Sur_centro = c(0, 1, -1, 0, 0),
   Occidente_Oriente = c(0, 0, 0, 1, -1)
            )) %>% data.frame()
```

	contrast	SE
Norte_sur	0.0029	0.0866
Sur_centro	-0.0287	0.0866
Occidente_Oriente	-0.0197	0.1016

¡Gracias!

Email: andres.gutierrez@cepal.org