Práctica: calibración

Máster de Análisis Político y Electoral (UC3M)

Nov. 2019

Situación inicial

En esta segunda fase de la práctica vas a construir un peso con las variables sociodemográficas para las que existen estimaciones poblacionales (`caut`, `tamuni`, `sexo`, `edad`, `ocupa` y `estud`).

Planificar la tarea

Recuerda los pasos necesarios para construir tu propio peso:

1. Generar un `vector` con nombres de **totales poblacionales**.
2. Preparar las **variables auxiliares** en la encuesta y el **peso inicial**.
3. Calcular los pesos mediante **calibración**.
4. **Evaluar** la calibración.
5. **Escalar** los pesos para que tengan media 1.

1. Datos y paquetes

Para este ejercicio vas a utilizar la encuesta `encuesta_gesop.RDS` y los datos poblacionales `datos_poblacionales.RDS` que se encuentran en la carpeta `datos`. Además, harás uso de los paquetes:

- `tidyverse`: contiene un conjunto de paquetes que facilitan la gestión y el análisis de los datos.
- `expss`: permite crear tablas personalizadas.
- `survey`: contiene las funciones `svydesign()` y `calibrate()` que permiten realizar la calibración.

1) **Cargar los paquetes**:

```
# install.packages("tidyverse")
# install.packages("survey")
# install.packages("expss")

library(tidyverse)
library(survey)
library(expss)
```

2) **Cargar los datos**:

```
encuesta <- read_rds("datos/encuesta_gesop.RDS")
datos_poblacionales <- read_rds("datos/datos_poblacionales.RDS")
```

2. Peso con las variables sociodemográficas

Vas a calibrar la muestra para que sea representativa con respecto a las variables sociodemográficas. Como ya viste en el análisis de representatividad, las variables `ocupa` y `estud` presentan desviaciones significativas. El resto de variables, como `sexo` o `edad`, también serán incluidas en la calibración para asegurar que no se producen desviaciones adicionales.

2.1 Crear un vector con los totales poblacionales

En primer lugar necesitas crear un vector con los totales poblacionales para que el modelo sepa cuál es la distribución poblacional de las variables en la muestra. Para ello vas a utilizar los `datos_poblacionales`, que es un *data frame* que contiene las estimaciones para las variables de interés.

Los `datos_poblacionales` están expresados en porcentaje, así que el primer paso es **transformarlo en totales**. El censo electoral de residentes en España de las elecciones del 28-A era de 34.581.472.

El segundo paso consiste en eliminar la primera categoría de cada variable, para que el modelo de calibración pueda converger. En `datos_poblacionales` la variable `valor_orden` presenta el orden de las categorías dentro de la variable.

```
vars_socdem <- c("caut", "tamuni", "sexo", "edad", "estud", "ocupa")

pobla_socdem <- datos_poblacionales %>%
  mutate(total_pobla = round(pobla/100*34581472, 0)) %>%
  filter(variable %in% vars_socdem) %>%
  filter(valor_orden != 1)
```

Para crear el vector de **totales poblacionales**, que es un *input* necesario de la función `calibrate()`, utilizarás las columna `total_pobla` que generaste en el paso anterior. Para indicar a qué categoría corresponde cada total, asigna un nombre a cada elemento del vector. Esos nombres serán el resultado de a unir las columnas `variable` y `valor` utilizando la función `paste0()`.

Por último en este paso, añade un primer valor al vector `totales_pobla` que corresponda con el total de elementos de la población (N). El nombre de este elemento será `(Intercept)`.

```
totales_pobla <- pobla_socdem$total_pobla
names(totales_pobla) <- paste0(pobla_socdem$variable, pobla_socdem$valor)
totales_pobla <- c("(Intercept)" =  34581472, totales_pobla)
totales_pobla
```

```
##                        (Intercept)                          cautAragón
##                           34581472                              968281
##                         cautAsturias                        cautBalears
##                               864537                              760792
##                         cautCanarias                      cautCantabria
##                              1556166                              449559
##                  cautCastilla y León            cautCastilla - la Mancha
##                              1936562                             1521585
##                        cautCatalunya            cautComunitat Valenciana
##                              5325547                             3527310
##                     cautExtremadura                          cautGalicia
##                               864537                              2213214
##                          cautMadrid                           cautMurcia
##                              4703080                             1002863
```

```
##                          cautNavarra                          cautPaís Vasco
##                               484141                                1694492
##                         cautLa Rioja                              cautCeuta
##                               242070                                  69163
##                          cautMelilla          tamuni10.001 a 100.000 hab.
##                                69163                               13348448
##       tamuni100.001 a 500.000 hab.            tamuniMás de 500.000 hab.
##                              8334135                               5498454
##                            sexoMujer                             edad30-44
##                             17878621                               8610787
##                            edad45-59                              edad>60
##                              9751975                              11342723
##                    estudObligatorios                 estudPosobligatorios
##                             13486774                               4046032
## estudPosobligatorios profesionales                  estudUniversitarios
##                              6466735                               8437879
##                          ocupaParado                         ocupaInactivo
##                              2593610                              15354174
```

2.2 Preparar los datos de la encuesta

Las variables de la encuesta ya están preparadas. Sin embargo, antes de comenzar con la calibración es necesario crear un **peso inicial**. Ese peso inicial sería el peso de selección si existiera. Como no es el caso, vas a crear un peso (`peso_pobla`) que, para cada elemento de la muestra, será el resultado de dividir el total poblacional (N) entre el número de casos en la muestra (n).

```
encuesta <- encuesta %>%
  mutate(peso_pobla = 34581472/nrow(.))
```

> *Escalar el peso inicial. El modelo de calibración converge más rápidamente si el peso inicial está escalado de forma que la suma de los pesos sea equivalente al total poblacional. En caso de que exista un peso de selección, también se puede escalar al total poblacional con una sencilla operación.*

2.3 Declarar el diseño de la encuesta y la calibración

Antes de proceder con el modelo de calibración hay que declarar el diseño de la encuesta utilizando `svydesign()` del paquete `survey`. Este paso sirve para detallar las características de la muestra como son:

- Los datos (`data`).
- La variable que sirve para identificar los conglomerados (`id`). El argumento `id = ~ 0` declara que la muestra no fue seleccionada a partir de conglomerados.
- Los pesos de selección o el peso inicial (`weights`).

> *El paquete* ***survey*** *permite hacer análisis de datos de encuesta. Además de la posibilidad de calcular los pesos por calibración, también se puede utilizar para realizar estimaciones teniendo en cuenta todos los elementos del diseño de la encuesta. Por ejemplo, la función* ***svymean()*** *permite calcular la media y el error estándar verdadero.*

```r
svy_des <- svydesign(id = ~ 0, weights = ~ peso_pobla, data = encuesta)
```

La **función `calibrate()` tiene cinco argumentos básicos**.

- En `design` hay que especificar el diseño de la encuesta del paso anterior.
- La `formula` siempre está precedida por `~` y contiene las variables que van a tomar parte en la calibración. Como side un modelo se tratara, las variables se separan por el operador `+`.
- El argumento `population` se corresponde con el vector de `totales_pobla` que creaste en el primer paso.
- `calfun` sirve para especificar el tipo de calibración a realizar. Hay tres tipos `"logit"`, `"linear"` y `"raking"`. La claibración logística impide que se generen pesos negativos, tal y como se especifica en el argumento `bounds`.
- En `bounds`, que es obligatorio si `calfun = "logit"`, se recogen los límites inferior y superior entre los que se deben de mover los pesos. Lo normal es establecer el cero y un valor relativamente alto.

```r
calib_socdem <- calibrate(design = svy_des,
                          formula = ~ caut + tamuni + sexo + edad + estud + ocupa,
                          population = totales_pobla,
                          calfun = "logit",
                          bounds = c(0, 999999))
```

2.4 Evaluar el peso

La evaluación del peso comprende tres fases:

1. Se **comprueba el peso en sí**, para identificar pesos extremos y hacer las transformaciones pertinentes.
2. Se comprueba que la calibración ha hecho su trabajo, comparando las **variables auxiliares con los totales poblacionales**.
3. Se evalúa el **efecto del peso en algunas de las variables de interés** para el análisis.

2.4.1 Estadísticos descriptivos del peso

El objeto `calib_socdem` es una lista que contiene información acerca del diseño de la encuesta. De aquí en adelante sólo necesitarás los pesos, así que para extraerlos puedes utilizar la función `weights()`. El resultado es un vector con tantos pesos como casos hay en la encuesta. Ese vector se incluye en la encuesta como una variable más.

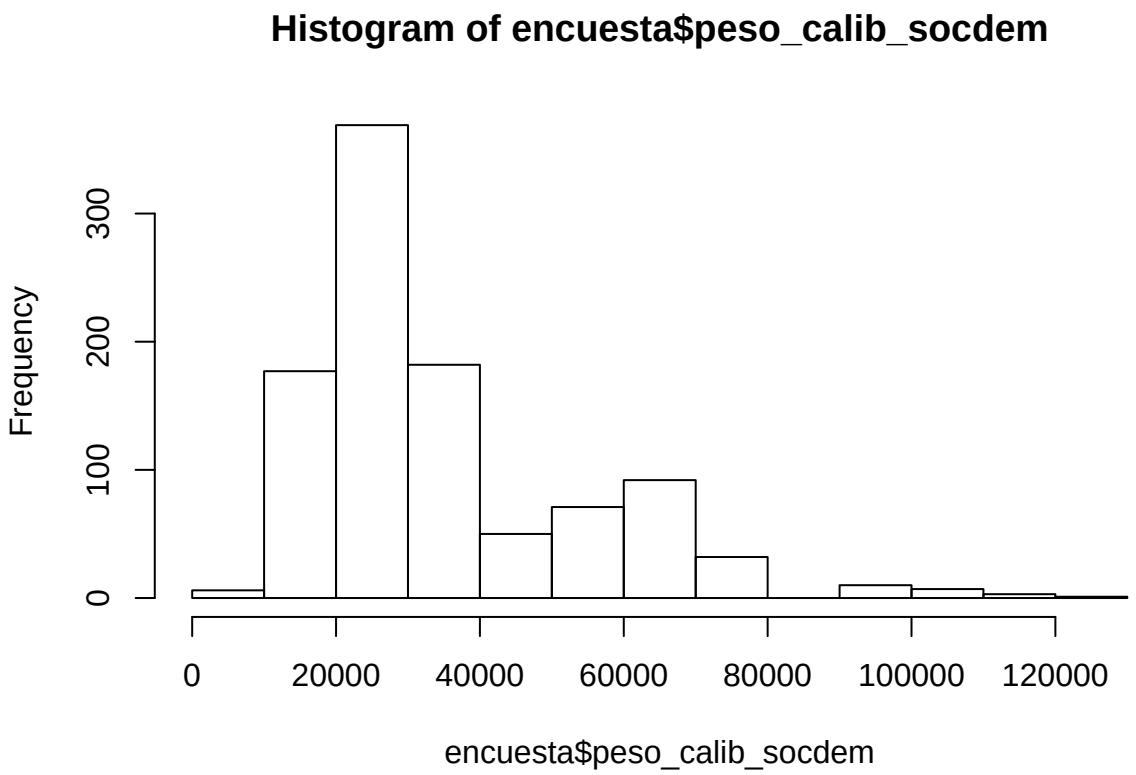
```r
peso_calib_socdem <- weights(calib_socdem)
encuesta$peso_calib_socdem <- peso_calib_socdem
```

El siguiente paso es generar una tabla de estadísticos descriptivos y un histograma del peso. El objetivo es evaluar la distribución y detectar valores anormales.

```r
sjmisc::descr(encuesta$peso_calib_socdem)
```

```
##
## ## Basic descriptive statistics
##
##  var    type label    n NA.prc     mean       sd     se       md  trimmed
##   dd numeric    dd 1000      0 34581.47 19645.74 621.25 25864.75 32027.46
##                          range skew
##  116121.4 (6423.34-122544.74) 1.37
```

```r
hist(encuesta$peso_calib_socdem)
```



Outliers y recortar pesos. En ocasiones los pesos producen valores extremos, lo que provocaría que las estimaciones sean menos precisas. Para detectar los pesos extremos hay que observar las distancias entre los pesos, aunque siempre depende del contexto. Una táctica es recortar los pesos al 99% o al 95%.

2.4.2 Comprobar que los totales poblacionales y la muestra ponderada coinciden

Ahora hay que crear una tabla de las variables empleadas en la calibración utilizando el peso. Para ello se utiliza el paqute `expss`, como en la práctica de anñalisis de representatividad.

```r
tabla <- encuesta %>%
  tab_weight(peso_calib_socdem) %>%
  tab_cells(caut, tamuni, sexo, edad, estud, ocupa) %>%
  tab_stat_cpct(total_statistic = "w_cpct") %>%
  tab_pivot()

colnames(tabla) <- c("variable_valor", "calibrado")
```

La tabla de la muestra ponderada se combina con los `datos_poblacionales` para crear una variable `dif` que sea la diferencia entre `calibrado` y `pobla`. La suma de la variable `dif` debería ser aproximadamente cero.

```r
tabla_checks <- left_join(tabla, datos_poblacionales, by = "variable_valor") %>%
  mutate(dif = round(calibrado - pobla, 1))
sum(tabla_checks$dif, na.rm = T)
```

```
## [1] 0.2
```

2.4.3 Comprobar el efecto del peso sobre las variables de intéres

Por último, se evalúa el efecto del peso en las variables de interés, en este caso `idv`. Para ello se comparan las estimaciones entre `idv` sin ponderar e `idv` ponderada.

```
sjmisc::frq(encuesta$idv)
```

```
##
## Intención directa de voto 28-A (P3) (x) <categorical>
## # total N=1000  valid N=1000  mean=9.33  sd=7.60
##
##           val frq raw.prc valid.prc cum.prc
##            PP 116    11.6      11.6    11.6
##          PSOE 213    21.3      21.3    32.9
##           C's  95     9.5       9.5    42.4
##            UP  47     4.7       4.7    47.1
##           JxC   6     0.6       0.6    47.7
##           ERC  45     4.5       4.5    52.2
##           ECP  13     1.3       1.3    53.5
##           PNV   8     0.8       0.8    54.3
##         Bildu   6     0.6       0.6    54.9
##      En Marea   8     0.8       0.8    55.7
##            CC   1     0.1       0.1    55.8
##     Compromís  10     1.0       1.0    56.8
##           Vox  88     8.8       8.8    65.6
##           CUP   0     0.0       0.0    65.6
##          Nulo  10     1.0       1.0    66.6
##  Otro partido  12     1.2       1.2    67.8
##     En blanco  16     1.6       1.6    69.4
##    No votaría  56     5.6       5.6    75.0
##            NS 167    16.7      16.7    91.7
##            NC  83     8.3       8.3   100.0
##          <NA>   0     0.0        NA      NA
```

```
sjmisc::frq(encuesta$idv, weights =  encuesta$peso_calib_socdem)
```

```
##
## Intención directa de voto 28-A (P3) (xw) <categorical>
## # total N=34581471  valid N=34581471  mean=9.46  sd=7.63
##
##           val     frq  label raw.prc valid.prc cum.prc
##            PP 3934595 <none>   11.38     11.38   11.38
##          PSOE 7704772 <none>   22.28     22.28   33.66
##           C's 3037021 <none>    8.78      8.78   42.44
##            UP 1433372 <none>    4.14      4.14   46.58
##           JxC  147670 <none>    0.43      0.43   47.01
##           ERC 1445533 <none>    4.18      4.18   51.19
##           ECP  431891 <none>    1.25      1.25   52.44
##           PNV  272304 <none>    0.79      0.79   53.23
##         Bildu  230458 <none>    0.67      0.67   53.89
```

```
##      En Marea  307058 <none>    0.89      0.89   54.78
##            CC   22871 <none>    0.07      0.07   54.85
##     Compromís  349008 <none>    1.01      1.01   55.86
##           Vox 3142363 <none>    9.09      9.09   64.94
##           CUP       0 <none>    0.00      0.00   64.94
##          Nulo  334244 <none>    0.97      0.97   65.91
##  Otro partido  322700 <none>    0.93      0.93   66.84
##     En blanco  599349 <none>    1.73      1.73   68.58
##    No votaría 1998655 <none>    5.78      5.78   74.36
##            NS 5944158 <none>   17.19     17.19   91.55
##            NC 2923449 <none>    8.45      8.45  100.00
##          <NA>       0     NA    0.00        NA      NA
```

2.5 Escalar y finalizar el peso

Por último, se escala el peso para que tenga media 1.

```
encuesta <- encuesta %>%
  mutate(peso_calib_socdem = peso_calib_socdem/mean(peso_calib_socdem))
sjmisc::descr(encuesta$peso_calib_socdem)
```

```
##
## ## Basic descriptive statistics
##
##  var    type label    n NA.prc mean   sd   se   md trimmed
##   dd numeric    dd 1000      0    1 0.57 0.02 0.75    0.93
##             range skew
##  3.36 (0.19-3.54) 1.37
```