Documento Metodológico

Casen 2013

Metodología actualizada de estimación para áreas pequeñas (SAE): Anexo de programación de la estimación de la tasa de pobreza por ingresos a nivel comunal (2011-2013)

Serie Documentos Metodológicos N°32
24 de junio de 2016

Ministerio de Desarrollo Social

Gobierno de Chile

www.desarrollosocial.cl

Observatorio Social

Contenido

1. Estimación para Áreas Pequeñas (SAE)

La metodología SAE aplicada por el Observatorio Social utiliza como base la Encuesta Casen y datos administrativos y censales[1]. En particular las estimaciones SAE se han realizado para la tasa de pobreza (2011 y 2013). El proceso para estas dos estimaciones tiene las mismas etapas, y a continuación se describen los pasos a seguir en la aplicación de la metodología.

Un procedimiento similar se llevó a cabo previamente, en el contexto de la difusión de resultados de la tasa de pobreza por ingresos a nivel comunal, basados en las encuestas Casen 2011 y 2009. Sin embargo, en enero 2015, el Ministerio de Desarrollo Social difundió una metodología actualizada de medición de pobreza por ingresos, para su aplicación a nivel nacional y regional, que dio pie a realizar una revisión al modelo de estimación sintético utilizado para el cálculo de la tasa de pobreza a nivel comunal. Los procedimientos a que refiere el presente informe son coherentes con tal metodología actualizada y permiten replicar los resultados obtenidos a nivel comunal para los años 2013 y 2011, con metodología SAE actualizada.

Paso 1: Suavización de factores de expansión.

Durante ejercicios previos se observó que algunos factores de expansión comunales tenían valores atípicos, con valores extremos como 0 y 1. Para solucionar esto se utilizó el método de Potter(2003) para suavizar los factores de expansión y de esta forma, evitar que *valores extremos* (outliers) en el factor de expansión influencien en forma negativa la contribución de la estimación directa a la tasa de pobreza de áreas pequeñas.

Paso 2: Estimación directa de la variable objetivo.

En el caso de pobreza en los años 2011 y 2013, se calcula la tasa de pobreza comunal a partir de la encuesta Casen, utilizando los factores de expansión suavizados; dichos factores suavizados son re escalados para mantener la consistencia con las estimaciones de población. Estas estimaciones corresponden a lo que se conoce como la estimación directa y es el primer componente de la estimación SAE.

Paso 3: Construcción de una base de datos a nivel comunal, para el vector de datos administrativos.

La base de datos administrativos considerada para generar el modelo de estimación fue la misma para ambos años y modelos. Esta base contiene variables del Censo 2002 tales como: porcentaje de analfabetos y porcentaje de población indígena. Por otro lado hay una gran cantidad de variables de registros administrativos que provienen de diferentes fuentes como el Ministerio de Educación (Mineduc), la Junta Nacional de Auxilio Escolar y Becas (Junaeb), el Servicio de Impuestos Internos (SII), la Superintendencia de Pensiones, así como el propio Ministerio de Desarrollo Social. Las variables del Censo de Población corresponden a las del año 2002; las variables administrativas, por otro lado, se actualizan en el 2011 y 2013 a una fecha que sea lo más cercana al levantamiento de Casen.

[1] Ver en Anexo una lista de las variables administrativas y censales.

Paso 4: Transformación de la variable objetivo para estabilizar la varianza muestral.

Uno de los supuestos del modelo de Fay-Herriot es el de varianza conocida y estable. Dado que la varianza de la variable objetivo no es conocida, es necesario estimarla; en segundo lugar, es necesario realizar ajustes para estabilizarla.

Para el caso de estimaciones de la tasa de pobreza en los años 2011 y 2013, la varianza no es constante ya que depende del valor de la tasa de pobreza[2]. Para corregir este problema, se aplica un procedimiento que permite estabilizar la varianza mediante la aplicación de la función arcoseno sobre la raíz cuadrada de P_i (Carter y Rolph 1974, pág. 882).

Una vez estabilizada la varianza, se realiza la estimación de ésta, tomando en consideración el diseño complejo de la Encuesta Casen. De esta forma la varianza de la estimación directa queda definida como:

$$D_i = Var(Y_i) \cong \frac{1}{4m_i} = \frac{deff_i}{4n_i}$$

En el modelo de Fay-Herriot se estima la varianza mediante una función generadora de varianza (FGV)[3] y al mismo tiempo se hace una transformación logarítmica para estabilizar la varianza. En el caso de Chile se trató de avanzar por esta línea, se hicieron múltiples pruebas para llegar a un buen modelo de FGV. Sin embargo, en el caso de Fay-Herriot la aproximación es bastante simple ya que usan una encuesta con muestreo aleatorio simple, mientras que la Encuesta Casen tiene un diseño complejo en el cual intervienen distintas variables que afectan la dispersión de las estimaciones[4]; luego de probar distintos modelos se comprobó que el ajuste de la estimación era muy bajo, por lo que no se lograría contar con un buen estimador de la varianza.

Para la estimación y estabilización de la varianza se siguió un camino alternativo sugerido por el experto internacional, Dr. Lahiri, el cual utiliza como estimador de la varianza la estimación directa de Casen y luego la estabiliza usando un procedimiento de "smooth variance"[5].

Como resultado de este procedimiento se tienen estimaciones suavizadas de la varianza de la estimación directa. Cabe destacar que estas estimaciones tienen un comportamiento poco regular, con valores muy extremos para algunas comunas muy pequeñas, en general, el resultado que se obtiene son varianzas superiores a las estimaciones de Casen, lo cual es esperable en comunas con bajo nivel de muestra, pero la magnitud es demasiado grande.

Paso 5: Selección del modelo de área y parámetros beta.

Con la base de datos comunales recopilada se estiman modelos de predicción para la variable objetivo. La elección del modelo debe buscar una buena predicción y no causalidad, por esto se

[2] La varianza de la estimación de tasa de pobreza corresponde a $\text{var}(P_i) = (P_i)(1-P_i)/(n_i - 1)$

[3] Una FGV es básicamente un modelo de regresión donde se trata de modelar el coeficiente de variación en función de variables asociadas al muestreo o diseño de la Encuesta.

[4] Segmentos, estratos, número de hogares en la muestra y otros.

[5] Para más detalle, ver minuta técnica enviada por experto internacional, Partha Lahiri (PhD).

evalúa el modelo en función del ajuste, analizando el R2 ajustado y en función de su simplicidad, utilizando criterios de información como el Akaike y Schwartz.

Paso 6: Estimación de parámetros relevantes: A, D y B.

El principal parámetro a estimar consiste en la varianza de la estimación sintética. Para el proceso de estimación de las tasas de pobreza se obtuvo la varianza A mediante una estimación en el programa R, que realiza una estimación conjunta de los parámetros beta y A, aplicando mínimos cuadrados ponderados.

Con la estimación del parámetro A, correspondiente a la varianza de la estimación sintética[6] y el parámetro D, referido a la varianza de la estimación directa, se calcula el ponderador B o "*shrinkage factor*" que combinará ambas estimaciones.

Paso 7: Cálculo de las estimaciones sintéticas de pobreza.

Una vez generado el modelo, se estiman los parámetros beta que serán utilizados para obtener la estimación sintética. En el proceso de cálculo de tasas de pobreza comunales, la selección del modelo se realizó con las comunas de mayor tamaño[7], debido a que comunas más grandes deberían presentar menores errores de muestreo en la variable dependiente (las variables independientes provienen de registros administrativos, por lo que no tienen un error de muestreo asociado).

Los coeficientes betas fueron estimados con esta selección de comunas y luego, para la estimación sintética, se utilizaron estos parámetros en todas las comunas.

Paso 8: Cálculo de las estimaciones Bayesianas de la tasa de pobreza.

Este paso es común a todas las estimaciones, y consiste en la ponderación de la estimación sintética y directa.

$$\hat{\Theta}_i^{EB} = (1 - \hat{B}_i)Y_i + \hat{B}_i Y_i^*$$

Paso 9: Truncamiento de la estimación Bayesiana.

Los estimadores Bayesianos pueden presentar un desempeño adecuado *en general*, pero un desempeño débil para algunos de los componentes *en particular*. Aplicado al contexto chileno, esto quiere decir que el modelamiento que beneficia a *la mayoría* de las comunas puede ser inapropiado para *algunas* comunas. Esta situación se puede dar, por ejemplo, por mala especificación del modelo, datos administrativos con error de medición distinto entre comunas, o por la presencia de valores extremos (outliers).

Para corregir por las potenciales fallas del estimador Bayesiano, se utiliza una corrección que trunca las estimaciones bayesianas en una banda de una desviación estándar en torno a las estimaciones directas (Efron y Morris, 1972; Fay y Herriot, 1979)

[6] En este documento el parámetro A corresponde a la varianza de la estimación sintética.
[7] Medido como comunas con más de 10 mil habitantes.

Paso 10: Transformación de las estimaciones Bayesianas de la tasa de pobreza a su escala original.

Es necesario recordar que para garantizar la estabilidad de la varianza, en este caso se aplica la función arcoseno de la raíz cuadrada a la variable pobreza, luego todas las estimaciones están en otra escala. Para obtener las estimaciones es la escala adecuada es necesario volver a transformar las tasas.

Paso 11: Cálculo de la "tasa de pobreza SAE", mediante la calibración de las estimaciones del nivel comunal al nivel regional.

Las estimaciones directas a nivel regional son confiables ya que el tamaño muestral asignado a cada región garantiza cierto nivel de precisión, por lo tanto, se espera que las estimaciones bayesianas del nivel comunal sean consistentes con la estimación regional correspondiente. Con esta finalidad, se realiza un ajuste final a las estimaciones bayesianas de tasa de pobreza comunal con el fin de imponer una consistencia lógica a los resultados. El ajuste, propuesto en Fay y Herriot (1979), consiste en hacer coincidir los niveles de pobreza regional obtenidos mediante la estimación Bayesiana con la estimación regional directa desde la encuesta.

La corrección consiste en estimar el número de pobres en cada región, sumando el número de pobres en cada comuna que pertenece a determinada región; el número de pobres se obtienen multiplicando la estimación de tasa de pobreza bayesiana por las proyecciones de población del INE. Este número se compara con el número de pobres que arroja la estimación regional.

La división entre la estimación regional y la suma de las estimaciones comunales es el factor de calibración que se calcula para cada región y se utiliza para calibrar las estimaciones comunales.

En el país hay 346 comunas incluyendo a la Antártica, sin embargo, no todas las comunas están representadas en la Encuesta Casen, y por lo tanto no siempre es posible obtener una estimación directa. El Ministerio ha abordado este problema utilizando un proceso de imputación de medias por conglomerados. De esta forma, el Ministerio obtiene estimaciones para 345 comunas en el país[8].

Es importante tener en cuenta que el proceso de calibración debe realizarse con todas las comunas del país, esto es así porque las estimaciones a nivel regional representan a todas las comunas[9]. Si sólo se hiciera con las comunas con muestra Casen y luego se agregaran las estimaciones para comunas sin muestra, entonces habría una inconsistencia entre el total regional y la suma de las comunas.

Paso 12: Cálculo de los intervalos de confianza de la tasa de pobreza SAE.

Una vez obtenidas las estimaciones puntuales a partir de aplicación de SAE, es necesario estimar los intervalos de confianza para el indicador de interés, ya que si bien con esta metodología se espera una reducción en los intervalos de confianza y mayor precisión, las cifras siguen siendo estimaciones y corresponde determinar su rango de variación.

[8] Para la comuna Antártica se asume una tasa de pobreza igual a cero.
[9] Estén o no en la muestra, esto se realiza a través del factor de expansión regional.

Los intervalos de confianza se estimaron mediante un proceso de bootstrap basado en Chatterjee, Lahiri y Li (2006). El proceso está detallado documentado en el documento "Procedimiento de cálculo de la Tasa de Pobreza a nivel Comunal mediante la aplicación de Metodología de Estimación para Áreas Pequeñas (SAE)" elaborado por el Ministerio.

El procedimiento de Bootstrap genera réplicas a partir de una distribución normal con parámetros que provienen de la estimación SAE.

Consideraciones:

Durante el procedimiento de estimación, y en particular para calcular la varianza del modelo, se realizó un proceso de optimización en R studio. En los pasos siguientes, se detalla la programación en Stata y en R respectivamente.

2. Imputación de medias por conglomerados

La encuesta Casen recopila información de 324 comunas, por lo que la metodología SAE sólo permite calcular el mismo número de tasas de pobreza. Esto genera un problema respecto de las comunas que no están presentes en Casen, por lo que este problema de información faltante impone la obligación[10] de ocupar otra estrategia para estimar la tasa de pobreza en comunas sin muestra Casen. El Ministerio utiliza un método de conglomerados (o clúster), mediante el cual identifica grupos de comunas con similares características (en adelante, clúster o conglomerados) en base a datos provenientes del Censo de Población y Vivienda. Realizada esta agrupación, es posible asignar a comunas sin representación en Casen, el promedio de la tasa de pobreza comunal del conglomerado al cual pertenecen. A continuación se detalla el procedimiento.

El problema de datos faltantes es común cuando se utilizan datos, ya sea proveniente de Encuestas o de registros administrativos. Las técnicas utilizadas para subsanar este problema son muy diversas y van desde eliminar las observaciones con datos faltantes, hasta métodos más sofisticados de imputación múltiple[11].

El Método de Conglomerados o Clúster aplicado actualmente por el Ministerio de Desarrollo Social, es el mismo utilizado para el cálculo del Indicador de Desarrollo Humano Comunal[12] del año 2000, método que resultó de un proceso conjunto de investigación realizado por el Ministerio y Programa de Naciones Unidas para el Desarrollo (PNUD). Este procedimiento ha sido utilizado además, en otros estudios de ambas instituciones y del Ministerio de Desarrollo Social con Unicef[13].

En líneas generales, este Método consiste en generar grupos de comunas en base a la similitud en ciertas variables seleccionadas. Una vez que se generan los grupos, se realiza una imputación de medias, asignando a las comunas sin dato, el dato promedio del grupo de comunas al cual pertenecen.

El primer paso en este método consiste en determinar las variables en base a las cuales se realizarán los grupos y la fuente de datos a utilizar. Con el objetivo de contar con información representativa a nivel nacional y que tuviera información completa para todas las comunas, se utilizó, en ausencia de una versión más actualizada, el Censo de Población y Vivienda 2002. Sólo la comuna Antártica ha sido omitida por considerarse que su situación es de distinta naturaleza al resto de las comunas del país.

[10] Según la Ley de Rentas Municipales (N°3.063), el Ministerio debe comunicar anualmente las estimaciones de pobreza para las 346 comunas del país con el objetivo de ser utilizadas como insumo en la asignación del Fondo Común Municipal. En este contexto, el Ministerio ha entregado oficialmente a la Subsecretaría de Desarrollo Regional estimaciones de tasas de pobreza comunal desde 2006.

[11] Ver "Imputación de datos: teoría y práctica" Medina y Galvan. Cepal 2007.

[12] "Desarrollo Humano en las Comunas de Chile", Mideplan- PNUD, 2000.

[13] "Las trayectorias del Desarrollo Humano en las comunas de Chile (1994-2003)", Mideplan-PNUD, 2005. "Índice de Infancia, una mirada comunal y regional", Mideplan-Unicef, 2002.

Las variables consideradas para la agrupación de comunas son: años promedio de escolaridad de adultos, población con trece y más años de estudios; y población sin escolaridad. Para seleccionar estas variables, se efectúan diversas pruebas de consistencia[14] previa, y una vez obtenidos los conglomerados, estos validan la selección de las variables, en el sentido de que el resultado sea coherente.

La generación de conglomerados se realiza en base a las 3 variables seleccionadas mediante el proceso de *"K-means"*, en el cual se le indica al programa computacional el número de conglomerados requerido[15] y el programa computacional determina automáticamente cuáles serán los centros de cada grupo, para posteriormente asignar cada comuna a un grupo particular en base a un criterio de distancia.

Como resultado de un proceso iterativo en que se prueban diferentes tamaños de conglomerado, se obtienen 28 grupos o conglomerados cumpliendo el criterio de que las comunas sean lo más similares posibles al interior de cada grupo (en las 3 variables seleccionadas) pero que a la vez, exista la mayor diferencia posible, en las variables seleccionadas, entre grupos distintos de comunas. En la selección de los grupos se exige que al menos una comuna en el grupo tenga información disponible para el dato de pobreza comunal proveniente de Casen.

Una vez generados los 28 grupos, se procede a calcular el promedio de las tasas de pobreza de las comunas en cada grupo. En primer lugar, se obtienen estimaciones directas del porcentaje de personas en condición de pobreza, en cada comuna con muestra Casen, utilizando la Encuesta Casen del año respectivo. A continuación a cada comuna con información faltante se le imputa el promedio de la tasa de pobreza (directa) del grupo al cual pertenece. Estas estimaciones basadas en imputación por conglomerados, realizadas para las comunas sin muestra Casen, son luego consideradas conjuntamente con las estimaciones de tasa de pobreza de comunas con muestra Casen (calculadas mediante metodología de estimación para áreas pequeñas, SAE[16]), para calibrar tales estimaciones a la incidencia de la pobreza a nivel regional. De esta forma, se obtienen estimaciones del porcentaje de personas en situación de pobreza para todas las comunas del país[17], que son consistentes con la medición regional de la tasa de pobreza.

De esta forma, el método de Conglomerados, que imputa la tasa de pobreza promedio de cada grupo de comunas a la(s) comuna(s) sin muestra Casen del mismo grupo, complementa las estimaciones realizadas por el Ministerio de Desarrollo Social, para comunas con muestra Casen, usando las estimaciones SAE realizadas. Con el objetivo de avanzar en la caracterización de la realidad social a nivel comunal, el Ministerio de Desarrollo Social se encuentra desarrollando una agenda permanente de investigación que permita extender y mejorar la aplicación de metodologías de estimación de aplicación local.

[14] Se realizan análisis exploratorios de funciones de distribución, matrices de correlación y poder explicativo de las variables.
[15] Como parte del análisis de conglomerados, se evalúan distintos números de conglomerados con el objetivo de mantener la mayor similitud posible al interior de los grupos y la mayor diferencia posible entre estos.
[16] Ver documento Procedimiento de cálculo de la Tasa de Pobreza a nivel Comunal mediante la aplicación de Metodología de Estimación para Áreas Pequeñas (SAE).
[17] Se generan datos para 345 comunas. Para la comuna Antártica se asume una tasa de pobreza de cero.

3. Utilización de la programación

Cada una de las programaciones descritas en los anexos atiende a los pasos mencionados previamente. Cabe destacar que el orden detallado previamente no necesariamente debe coincidir con el orden de las programaciones en el anexo.

Lo primero es “0_creacion_base_mini”. Esto permite trabajar una base de datos a nivel comunal, lo que es necesario para poder calcular la pobreza comunal. Luego, para poder suavizar los factores de expansión es necesario ejecutar la programación “k óptimo” y “MatrizK_30grupos”. De este modo, se obtienen estimaciones con un menor error cuadrático medio, ya que suavizar los factores de expansión reduce considerablemente la varianza del error (Paso 1).

Con los factores suavizados es posible calcular la pobreza que proviene directamente de Casen, y para eso es necesario usar “Estimaciones_pob_Wtrimm” (Paso 2).

Por otra parte, paralelamente existe una base de datos con los registros administrativos que hay que considerar necesariamente para realizar las estimaciones sintéticas (Paso 3).

Para cumplir con el paso 4 es necesario ejecutar la programación “Auxiliar” (que genera variables auxiliares para poder calcular el efecto diseño), y luego aplicar el programa “DEFF”, que utiliza las variables creadas anteriormente para considerar el efecto diseño en la estimación.

Luego, ejecute “Genera base 2”. Esta programación permite escoger el mejor modelo de un set de variables provenientes de registros administrativos (Paso 5).

Los pasos 6, 7, 8, 9, 10 y 11 Se concretan utilizando la programación “Estimation”. Este do file toma las variables escogidas en el paso anterior y realiza las estimaciones de pobreza sintética para finalmente calcular la pobreza bayesiana. Las tasas de pobreza que se salgan del rango de una desviación estándar serán truncadas, y luego es necesario realizar la transformación hacia atrás de la tasa de pobreza (recuerde que el proceso requiere realizar una transformación monotónica de la tasa de pobreza).

Finalmente el paso 12 puede ser obtenido a través del “Bootstrap”. Esto permite calcular los intervalos de confianza de las tasas de pobreza, y así mismo verificar que hay una mejora en términos de estimación, puesto que el intervalo de confianza obtenido es considerablemente menor que el intervalo obtenido solamente con información de la encuesta.

4. Programación de Do files paso a paso.

Esta programación toma la base de datos de Casen 2013, genera una variable dicotómica que toma el valor 1 si el individuo es pobre y 0 en otro caso, y guarda una base de datos con esta variable en conjunto con otras variables ya presentes en la encuesta que son relevantes.

1. 0_creacion_base_mini

```
*Genera Base Casen_mini2013*

global path1 "especificar ruta de archivo donde se ubican los insumos"
clear all
use "$path1\casen_identif_viv.dta"

gen poor_mn=.
replace poor_mn=0 if pobreza_mn==3
replace poor_mn=1 if (pobreza_mn==1 | pobreza_mn==2)


keep  folio o region comuna zona  varstrat varunit expr expc numper pco1 viv segmento sexo edad e1 o15 o17 ///
r6 r11a r11b r6 r13a r13b r13c r13d r13e r13f asiste esc activ oficio1 rama1 ytrabajocorh yoprcorh yautcorh ysubh
ymonecorh yaimcorh ytotcorh ///
hacinamiento pobreza_mn poor_mn
format folio %12.0f
save "$path1\casen_mini_1junio.dta", replace
```

Generación de población a nivel comunal y regional[18]

```
clear all
global path1 "especificar ruta de archivo donde se ubican los insumos"
global path2 "especificar ruta de archivo donde se ubican las bases principales"

use "$path2\demograficas ine 2011-2013.dta", clear

gen r=int(com_id/1000)

gen pobc=pobc2013
bysort r: egen pobreg_2013=sum(pobc)

bysort com_id: gen orden=_n
keep if orden==1
sort com_id

keep com_id pobreg_2013 pobc2013
save "$path1\pobreg_2013.dta", replace
```

Generación de población a nivel comunal y regional (de Casen)[19]

18 Programación utilizada para calibrar.
19 Programación utilizada para seleccionar las comunas grandes de Casen.

```
clear all
use "C:\Users\dvasquez\Desktop\Casen 2011\casen_2013_mn_b_principal.dta"
global path1 “especificar ruta de archivo donde se ubican los insumos”
*size_samp
egen co=group (comuna)
bys comuna: egen size_samp=count(co)

*size_pop
bys comuna: egen size_pop_c=sum(expc)
bys region: egen size_pop_r=sum(expr)


bysort comuna: gen orden=_n
keep if orden==1
gen com_id=comuna
sort com_id
keep comuna com_id size_samp size_pop_c size_pop_r

save "$path1\tamañomuestral", replace
```

2. K óptimo

Esta programación es parte del truncamiento de los factores de expansión. El hecho de truncar los factores significa incurrir en un futuro estimador sesgado, sin embargo el sesgo se ve compensado por una reducción aún mayor en los niveles de varianza, logrando finalmente un menor Error Cuadrático Medio, que es precisamente la ganancia de trabajar con este método. Principalmente, estas líneas calculan 40 Errores Cuadráticos Medios (número sugerido en literatura previa) para 30 grupos distintos (15 regiones y 2 zonas).

```
set more off
global path1 “especificar ruta de archivo donde se ubican los insumos”
global path2 “especificar ruta de archivo donde se ubican los resultados”
forvalues j = 1(1)30{
clear
clear matrix
set mem 300m
* base de datos CASEN (página web)
use "$path1\casen_mini_1junio.dta", clear
* definición de grupos relevantes
egen rezo=group(region zona)
table rezo [w=expc], c(m poor_mn)
keep if rezo==`j'
* pobreza CASEN por grupo
egen p=wtmean(poor_mn), weight(expc)
*pobreza casen por grupo
set more off
matrix drop _all
matrix MSE_`j'=J(40,4,.)
forvalues i = 1(1)40{
* aplicación del modelo potter (truncamiento de factores de expansión)
```

```
gen unos=1
egen ene=sum(unos)

gen hhh=expc*expc
egen ss1=sum(hhh)
gen ss2=ss1/ene
gen ss3=(`i'*ss2)^0.5
sum ss3
return list
matrix MSE_`j'[`i',3]=r(mean)
/* nuevo peso suavizado */
gen exprx=expc
replace exprx=ss3 if expc>ss3
/* ajuste a población de referencia */
egen norig=sum(expc)
egen npobregx=sum(exprx)
gen wx=norig/npobregx
gen wkx=wx*exprx
sum wkx
/* estimaciones de tasas de pobreza con factores suavizados */
svyset  [pw=wkx], psu(varunit) strata(varstrat) singleunit(centered)
svy: mean poor_mn
/* estimación del MSE */
matrix dd=e(b)
gen dd= dd[1,1]
matrix MSE_`j'[`i',4]=dd[1,1]
matrix var=e(V)
gen ee= var[1,1]

en mse=ee+(p-dd)^2
sum mse
matrix MSE_`j'[`i',1]=r(mean)
matrix MSE_`j'[`i',2]=`i'
drop unos-mse
}

matrix list MSE_`j'
svmat MSE_`j'
*OUTPUT: BASE DE DATOS CON 40 MSE POR GRUPO
save "$path2\grupo_`j'.dta", replace
}
```

3. MatrizK_30grupos

Esta programación elige el menor Error Cuadrático Medio en cada grupo calculado en la programación anterior, grabando estos K óptimos por grupo.

```
set more off
set mem 500m
global path1 “especificar ruta de archivo donde se ubican los insumos”
global path2 “especificar ruta de archivo donde se ubican los resultados”
```

```
matrix K=J(30,2,.)
forvalues i = 1(1)30{
clear
* INPUT: BASE DE DATOS FINAL DO FILE 1
use "$path2\grupo_`i'.dta", clear
* busca el MSE minimo en cada grupo
egen mse_min=min(MSE_`i'1)
gen resta=(MSE_`i'1-mse_min)
sum MSE_`i'3 if resta==0
return list
matrix K[`i',1]=r(mean)
matrix K[`i',2]=`i'
}
matrix colnames K= K grupo
matrix list K
svmat K
keep  K1 K2
ren  K2 grupo
sort grupo
* OUTPUT: BASE DE DATOS CON K OPTIMO POR GRUPO
save "$path2\K por grupo.dta", replace
```

4. Estimaciones_pob_Wtrimm

Esta programación permite calcular, utilizando factores de expansión suavizados (los cuales se truncaron utilizando los ECM (errores cuadráticos medios) calculados en los programas anteriores, la pobreza directa regional (llámese directa a que proviene de la misma encuesta) y la pobreza directa comunal. Hay que destacar que es similar a la estimación de pobreza que se calcula directamente de la encuesta, sin embargo, dado que se modifican los factores de expansión, difiere en cierta medida.

```
clear all
set mem 300m
set more off

global path1 “especificar ruta de archivo donde se ubican los insumos”
global path2 “especificar ruta de archivo donde se ubican los resultados”


use "$path1\casen_mini_1junio.dta", clear

gen com_id=comuna

* K OPTIMO PARA CADA GRUPO (minimiza MSE)
egen grupo=group(region zona)

sort grupo
merge grupo using  "$path2\K por grupo.dta"
keep if _merge==3
```

```
* nuevo peso suavizado: trunca en K optimo
gen exprx=expc
replace exprx=K if expc>K

* calibración del nuevo peso a la pobl original
egen norig=sum(expc)
egen npobregx=sum(exprx)

gen wx=norig/npobregx
gen wkx=wx*exprx

sum wkx

drop _merge
save "$path1\POBREZA ORIGINAL Pesos trunc.dta", replace

* NUEVA TASA DE POBREZA CON FACTORES SUAVIZADOS

svyset  [pw=wkx], psu(varunit) strata(varstrat) singleunit(centered)

* PARAMETROS REGIONALES: media y varianza
* se ocupa para estimar el efecto diseño

matrix PR=J(15,3,.)

forvalues i = 1(1)15 {
svy: mean poor_mn if region==`i'

matrix media_`i'=e(b)
matrix PR[`i',1]=media_`i'[1,1]

matrix var_`i'=e(V)
matrix PR[`i',2]=var_`i'[1,1]

sum region if region==`i'
matrix PR[`i',3]=r(mean)
}
matrix list PR

* PARAMETROS COMUNALES: media y varianza
* pobreza comunal estimada con nuevo factor suavizado

sort com_id
egen idc=group(com_id)
sum idc
global ncom=r(max)

matrix PC=J($ncom,3,.)

forvalues i = 1(1)$ncom {
svy: mean poor_mn if idc==`i'
```

```
matrix mediac_`i'=e(b)
matrix PC[`i',1]=mediac_`i'[1,1]

matrix varc_`i'=e(V)
matrix PC[`i',2]=varc_`i'[1,1]

sum com_id if idc==`i'
matrix PC[`i',3]=r(mean)
}
matrix list PC

** OUTPUT: BASE DE DATOS REGIONAL
preserve
drop region
svmat PR
ren PR1 media_PT
ren PR2 varianza_PT
ren PR3 region
keep media_PT varianza_PT region
drop if region==.
sort region
save "$path1\pregion_PT.dta", replace
restore

** OUTPUT: BASE DE DATOS COMUNAL
preserve
drop com_id
svmat PC
ren PC1 mediaCOM_PT
ren PC2 varianzaCOM_PT
ren PC3 com_id
keep mediaCOM_PT varianzaCOM_PT com_id
drop if com_id==.
sort com_id
save "$path1\pcomuna_PT.dta", replace
restore
```

5. Auxiliar

Esta programación genera variables que contienen un número de observaciones de personas y viviendas por comuna y región, necesarias para poder calcular el efecto diseño de la encuesta. El efecto diseño juega un papel fundamental en el cálculo de la varianza del modelo.

```
clear all
set mem 300m
set more off

global path1 “especificar ruta de archivo donde se ubican los insumos”
global path2 “especificar ruta de archivo donde se ubican los resultados”
```

```
use "$path1\casen_mini_1junio.dta", clear

gen com_id=comuna

gen unos=1

* numero de personas (muestral y expandido)
egen idsample_c=sum(unos) , by(comuna)
egen idsample_r=sum(unos) , by(region)

egen idexpan_c=sum(expc) , by(comuna)
egen idexpan_r=sum(expr) , by(region)

* numero de viviendas (muestral y expandido)
bysort region : gen idr=_n
bysort com_id : gen idc=_n

egen idviv2= group( region comuna zona segmento viv)
bysort  idviv2: gen idh=_n

egen hh_sample_c=count(unos) if idh==1, by(comuna)
egen hhsample_c=max(hh_sample_c), by(comuna)

egen hh_sample_r=count(unos) if idh==1 , by(region)
egen hhsample_r=max(hh_sample_r), by(region)

* guarda datos relevantes a nivel regional
preserve
keep if idr==1

sort region
drop if region==.
keep region idsample_r idexpan_r hhsample_r idsample_c
save "$path1\aux_region.dta", replace

restore
```

6. DEFF

Como se explica anteriormente, es necesario considerar el efecto diseño para poder estimar la varianza del modelo, y el cálculo de este efecto diseño se encuentra programado en las siguientes líneas.

```
clear all
set mem 300m
set more off

global path1 “especificar ruta de archivo donde se ubican los insumos”
global path2 “especificar ruta de archivo donde se ubican los resultados”
```

```
* base con pobreza estimada con factores suavizados *
use "$path1\pregion_PT.dta"
sort region
drop if region==.

* base con variables auxiliares a nivel regional *
merge region using "$path1\aux_region.dta"

* ESTIMACIÓN DEL EFECTO DISEÑO:
* se utiliza la pobreza y varianza regional estimada con factores suavizados

gen pq=(media_PT*(1- media_PT))
gen var_uw=pq/hhsample_r
gen deff_v= varianza_PT/var_uw

table region, contents(mean deff_v)

keep deff_v region hhsample_r
sort region

* OUTPUT: BASE DE DATOS CON EFECTOS DISEÑO POR REGION
save "$path1\DEFF.dta", replace
```

7. Genera Base 2

Esta programación prepara la información necesaria para realizar la estimación SAE. Se toman los datos de registros administrativos y los datos de pobreza comunal, así como el efecto diseño calculado. En particular, dado que la estimación lo requiere, se realiza una transformación monotónica de las variables que van a interactuar en el modelo. Estas líneas se encargan finalmente de preparar 3 bases de datos: Datos con las comunas de mayor tamaño, comunas de menor tamaño, y comunas que no están presentes en la encuesta Casen.

```
clear all
set dp comma
set more off
global path1 “especificar ruta de archivo donde se ubican los insumos”
global path2 “especificar ruta de archivo donde se ubican los resultados”
* BASE MIDEPLAN COMUNAL CON DATOS ADMINISTRATIVOS

*De registros administrativos
import excel “especificar ruta de archivo donde se ubican los insumos”, sheet("Sheet1") firstrow
*drop pobm2013 ad2013 nna2013

*Base de datos con efecto diseño
sort region
merge region using "$path1\DEFF.dta"
drop _m

*De casen
*Pobreza por comuna, factores corregidos
```

```
sort com_id
merge com_id using "$path1\pcomuna_PT.dta"
drop _merge

*Tamaño muestral
sort com_id
merge com_id using "$path1\tamañomuestral.dta"
drop _merge

*drop *2004 *2005 *2006 *2007 *2008 *2009 *2010 *2011

*calculando poblacion regional
preserve
egen size_popregion=sum(size_pop_r), by(region)
keep com_id size_popregion
sort com_id
save "$path1\pob_regional.dta", replace
restore

*borra antártica
drop if com_id==12202
drop if region==.
save "$path1\BASE_DATOS COMUNA.dta", replace

*ESTIMACIÓN DE POBREZA PREVIA PARA COMUNAS NO CASEN
tab  region, gen(reg_)

* DEFINICIÓN DE TASAS DE POBREZA REGIONALES CASEN NO TRUNCADAS
* fuente: CASEN 2011, mideplan
* table region [w=expr_r2], c(m poor)
* se ocupa para hacer el raking*/

gen zz=.
replace zz= 0.0822711 if region==1
replace zz= 0.0397128 if region==2
replace zz= 0.072631 if region==3
replace zz= 0.1623281 if region==4
replace zz= 0.1558895 if region==5
replace zz= 0.160127 if region==6
replace zz= 0.2231314 if region==7
replace zz= 0.2232621 if region==8
replace zz= 0.2790496 if region==9
replace zz= 0.1762216 if region==10
replace zz= 0.0677334 if region==11
replace zz= 0.0557527 if region==12
replace zz= 0.0917109 if region==13
replace zz= 0.2313524 if region==14
replace zz= 0.1457956 if region==15

preserve
bysort region: gen idr=_n
```

```
keep if idr==1
keep region zz com_id /*com_id es agregada, si no, los resultados no se podrían pegar después*/
ren zz  pob_orig_REG
sort region com_id
save "$path1\pobreza original regional SIN TRUNCAR.dta", replace
restore

*****************************************************************
* PREPARACION BASE DE DATOS
* TRANSFORMACION DE VARIABLES DEPENDIENTE E INDEPENDIENTE
* variables continuas: ln
* variables discretas: asin
*****************************************************************
sum

* arcoseno variable dependiente
*******************************
* a nivel comunal *

rename mediaCOM_PT pc
rename varianzaCOM_PT var_pc

gen raiz_pc=pc^0.5
gen asin_pc=asin(raiz_pc)

sum asin_pc

* a nivel regional, directas no truncadas, para raking *
gen raizREG_pc=zz^0.5
gen POVREG_ARC=asin(raizREG_pc)

*asalariados
gen k5=asin(sqrt(bsm2013))  /*bajo el salario mínimo*/

*del censo
gen c1=asin(sqrt(analf/100))
gen c4=asin(sqrt(tasa_etnia))

*dummies regionales
gen x1    =(region==        1        )
gen x2    =(region==        2        )
gen x3    =(region==        3        )
gen x4    =(region==        4        )
gen x5    =(region==        5        )
gen x6    =(region==        6        )
gen x7    =(region==        7        )
gen x8    =(region==        8        )
gen x9    =(region==        9        )
gen x10   =(region==        10       )
gen x11   =(region==        11       )
gen x12   =(region==        12       )
```

```
gen x13   =(region==          13          )
gen x14   =(region==          14          )
gen x15   =(region==          15          )

*fonasa
gen f1=asin(sqrt(prop_fonasa_a))
replace prop_fonasa_ab=1 if prop_fonasa_ab>1
gen f2=asin(sqrt(prop_fonasa_ab))

*isapre
gen i1=asin(sqrt(prop_isapre))

sum size_pop_c, d

* estimación varianza D conocida
* a nivel comunal
* Yi/teta_i ~ N(teta_i, tdor_i)

*********************************
table region, contents(mean deff )

gen tdor=(deff_v/(4*size_samp))
sum tdor

* redefinición de variables
gen d=tdor
gen y=asin_pc

regress asin_pc k5 c1 c4 f2 i1 x4 x5 x8 x9 x14 if size_pop_c>10000
estimate store e1
outreg2 [e1] using "$path2\modelos_sae.xls", append

*********************************
* comunas sin dato CASEN
preserve
keep if y==.
*gen comuna=com_id
sort comuna
save "$path1\SIN POV VAR.dta", replace
restore
*********************************

* se borran comunas sin datos claves: Y, D
drop if y==.
drop if d==.

*********************************
preserve
sort com_id
save "$path1\BASE_SAE.dta", replace
restore
```

```
*******************************

*******************************
preserve
keep if size_pop_c>10000
save "$path1\BASE.dta", replace
restore
*******************************

*******************************
preserve
keep if size_pop_c<=10000
save "$path1\BASE_aux.dta", replace
restore
*******************************
```

8. Estimation

Esta programación estima los datos de pobreza SAE. Para ello, se toma la base anterior, que contiene datos de la encuesta y datos de registros administrativos, y en R se calcula la varianza del modelo. Con esto, es posible calcular la estimación bayesiana de pobreza.

Además, se truncan las estimaciones que superen en una desviación estándar al valor puntual (independiente de la cola), de tal forma que la estimación no afecte en demasía el valor encontrado desde la misma encuesta.

```
clear all
set dp comma
set more off
global path1 "especificar ruta de archivo donde se ubican los insumos"
global path2 "especificar ruta de archivo donde se ubican los resultados"

* 1. Definición de variables X's usadas en regresión ponderada
*****************************************************************

global vardep1  f2 i1 x9 c4 x5 x14 k5 x8 c1 x4
global vardep2  f2, i1, x9, c4, x5, x14, k5, x8, c1, x4

use "$path1\BASE.dta", clear

*drop m* p*

mkmat y $vardep1 d

scalar m = rowsof(y)

matrix cte=J(m,1,1)

* MATRIX OF X's
```

```
matrix X=cte,$vardep2

scalar p=colsof(X)

di m
di p

saveold "$path1\base comunal v2.dta", replace

* 2. RUN R FROM STATA
* solution of A using REML method
* comunas >10.000 hbts
*************************************************************************

matrix A_REML=J(1,1,.)

preserve

set more off

global path1 "especificar ruta de archivo donde se ubican los insumos"

log using "$path1\A_remlprueba.log", replace

global Rterm_path `"especificar ruta de archivo donde se ubica R studio"'
global Rterm_options `"--vanilla"'

rsource using "$path1\ESTIMACION A_REML EN R", lsource
```

ESTA ES LA PROGRAMACIÓN EN R STUDIO:

```
# Application of general Fay-Herriot model using Saipe data for the year 1993
# unbalanced case i.e. each small area has unequal no of obs
# unequal sampling variance is considered for all areas

library(foreign, pos=4)
Datos <- read.dta("especificar ruta de archivo donde se ubican los insumos"/base comunal v2.dta",
convert.dates=TRUE, convert.factors=FALSE, missing.type=FALSE, convert.underscore=FALSE,
warn.missing.labels=TRUE)
attach(Datos)
names(Datos)
#summary(Datos)
m = nrow(Datos)

library(MASS)

X = cbind(rep(1,m),f2, i1, x9, c4, x5, x14, k5, x8, c1, x4)
p = ncol(X)

################################## Empirical Bayes ##############################
```

```
### solution of a.hat using REML method

library(MASS)
reml.soln.a = function(a.hat){
w.a = diag(1/(a.hat + d))
sigma.a = t(X) %*% w.a %*% X
inv.sig.a = solve(sigma.a)
p.a = w.a - w.a %*% X %*% inv.sig.a %*% t(X) %*% w.a
tr.p.a = sum(diag(p.a))
tr.w.a = sum(diag(w.a))

#return(0.5*((t(y) %*% p.a %*% p.a %*% y) - tr.p.a))
return(0.5*((t(y) %*% p.a %*% p.a %*% y)-(tr.w.a))+(1/a.hat))
}


a.reml= uniroot(reml.soln.a, c(-20,20))$root

if (a.reml< 0) {a.reml = 0} else
{a.reml = a.reml}
a.reml

warnings()

log close

clear
insheet using "$path1\A_remlprueba.log"

split v1, generate(e) parse(])

gen a_reml=real(e2)
sum a_reml

matrix A_REML[1,1]=r(mean)

restore

matrix list A_REML
scalar a_reml=A_REML[1,1]
di a_reml

* 3. EMPIRICAL BAYES
* EB estimate of theta using REML estimate of A(variance component)
*********************************************************************
gen  b_reml=(d/(a_reml+d))
mkmat b_reml
matrix list b_reml

gen aux1=1/(a_reml+d)
```

```
mkmat aux1
matrix w_a_reml=diag(aux1)

matrix beta_hat_reml=inv(X'*w_a_reml*X)*X'*w_a_reml*y /*no es igual que la regresión lineal de las variables*/
matrix list beta_hat_reml

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
* se completa la base de datos con comunas con <10.000 hbts
* se completa la base de datos con comunas NO CASEN
* total 345 comunas

matrix drop X y cte d
scalar drop p m
matrix drop b_reml w_a_reml
drop b_reml aux1

gen orig=1

* comunas con menos de 10.000 hbts
append using "$path1\BASE_aux.dta"
gen orig2=1

* comunas sin pobreza directa
append using "$path1\SIN POV VAR.dta"
replace d=. if y==.

gen orig3=1

mkmat y $vardep1 d
scalar m = rowsof(y)
matrix cte=J(m,1,1)
matrix X=cte,$vardep2
scalar p=colsof(X)

sum y $vardep1 d

* estimación B: 345 comunas
gen  b_reml=(d/(a_reml+d))
mkmat b_reml

gen aux1=1/(a_reml+d)
mkmat aux1
matrix w_a_reml=diag(aux1)
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

* estimador sintético: 345 comunas
matrix synth_reml=X*beta_hat_reml
matrix list synth_reml
svmat b_reml
svmat synth_reml
```

```
replace synth_reml=0 if synth_reml<0                    /*hay comunas sin pobreza directa que tienen predicho
pob=0*/

sum synth_reml
sum synth_reml if orig==1
sum synth_reml if orig2==1
sum synth_reml if orig2==.

* estimación de theta EB
gen theta_reml=(1-b_reml)*y+b_reml*synth_reml
replace theta_reml=synth_reml if orig2==.               /*se imputa la predicción XB a comunas sin pobreza directa*/

gen p_predicho1=theta_reml
gen p_pred=p_predicho1

gen poor_cal=p_pred

sum y synth_reml p_predicho
sum y synth_reml p_predicho if orig==1
sum y synth_reml p_predicho if orig2==.

* 4. LIMITED TRANSLATION METHOD
*************************************************************************
gen limit_1=(y-(d^0.5)*1)
gen limit_2=(y+(d^0.5)*1)

gen marca=0
replace marca=1  if poor_cal<limit_1 & orig2!=.
replace marca=2  if poor_cal>limit_2 & orig2!=.

tab marca

preserve
sum limit_1 poor_cal limit_2 if marca==1
sum limit_1 poor_cal limit_2 if marca==2

gen poor_sae=poor_cal if limit_1<=poor_cal & poor_cal<=limit_2 & orig2!=.
replace poor_sae=limit_1  if poor_cal<limit_1 & orig2!=.
replace poor_sae=limit_2  if poor_cal>limit_2 & orig2!=.

replace poor_cal=sin(poor_cal)^2
replace y=sin(y)^2
replace limit_1=sin(limit_1)^2
replace limit_2=sin(limit_2)^2
replace synth_reml =sin(synth_reml)^2
replace theta_reml=sin(theta_reml)^2
replace poor_sae=sin(poor_sae)^2

list com_id comuna limit_1 poor_cal limit_2 if marca==1
list com_id comuna limit_1 poor_cal limit_2 if marca==2
```

```
*SOLO COMUNAS CASEN

drop if y==.
sort poor_cal
gen idk=_n
egen a_1=mean(limit_1), by(idk)
egen a_2=mean(limit_2), by(idk)

twoway (line poor_cal idk , sort) ///
(line a_1 idk , sort lpattern(dash) lcolor(ltblue)) ///
(line a_2 idk , sort lpattern(dash) lcolor(ltblue)) ///
(scatter a_1 idk  if marca==1, sort mcolor(black) msymbol(circle_hollow)) ///
(scatter a_2 idk  if marca==2, sort mcolor(black) msymbol(lgx)) , ///
legend(order(1 "tasa de pobreza bayesiana" 2 "limite inferior" 3 "limite superior" 4 "pobreza bayesiana > limite" 5
"pobreza pobreza bayesiana < limite")) xlabel(minmax) xtitle(comunas (ordenadas por tasa de pobreza)) ///
graphregion(fcolor(white) lcolor(white) ifcolor(white) ilcolor(white)) title(Modelo nuevo)
restore

replace poor_cal=poor_cal if limit_1<=poor_cal & poor_cal<=limit_2 & orig2!=.
replace poor_cal=limit_1  if poor_cal<limit_1 & orig2!=.
replace poor_cal=limit_2  if poor_cal>limit_2 & orig2!=.

drop C D E F G H I J K L M N O P Q R

* stored relevant variables (important for the bootstrap procedure)
*************************************************************************
preserve
drop if orig2==.

matrix VA=A_REML[1,1]
svmat VA
egen a_reml=max(VA)

gen A=a_reml
gen B=b_reml
gen dsd=d^0.5
gen F=(1-B)^0.5

keep com_id poor_cal d a_reml b_reml F dsd size_samp  size_pop* region marca
sort com_id
save "$path1\theta fu8ll sample.dta", replace
restore

preserve
keep if orig2==.

matrix VA=A_REML[1,1]
svmat VA
egen a_reml=max(VA)

keep com_id poor_cal a_reml size_samp  size_pop* region comuna
```

```
sort com_id
save "$path1\theta fu8ll sample_NOCASEN.dta", replace
restore

* 5. BACKTRANSFORMATION
*************************************************************************
gen pobreza=poor_cal
replace pobreza=sin(pobreza)^2

* 6. RAKING, CALIBRATION OF POVERTY RATE
*************************************************************************
*drop size_pop
sort com_id

*agregando pobreza de comunas sinmuestra CASEN por método clúster
merge 1:1 com_id using "$path1\pobreza no casen 2013.dta"
tab _merge
replace pobreza=tasan if pobreza==.
drop _merge tasan

* tamaño poblacional comunal (según proyección INE)
sort com_id
merge com_id using "$path1\pobreg_2013.dta"
tab _m
keep if _m==3
drop _merge

* tasas de pobreza regional directas sin truncar
sort region
merge region using "$path1\pobreza original regional SIN TRUNCAR.dta"
tab _m
drop _m

bysort comuna: gen idc=_n
bysort region: gen idr=_n

* número original de pobres (a nivel regional)
gen nreg_orig=(pob_orig_REG*pobreg_2013) /*pobreza regional pesos sin truncar*/

* número de pobres predichos (nivel comunal)
gen ncom_pred=(pobreza*pobc2013)
egen n_comp=sum(ncom_pred) if idc==1, by(region)
egen ncomp=max(n_comp), by(region)

* razón
gen cal=nreg_orig/ncomp

sum cal
tabstat cal, statistics( mean ) by(region) columns(variables)

di a_reml
```

```
* ranking de tasa de pobreza
replace pobreza=pobreza*cal

sum pobreza pob_orig_REG
*tomar base casen, calcular pobreza y pegarle esta
save "$path1\POBREZA_FINAL_SAE.dta", replace

preserve
drop if orig2==.
keep com_id  y cal synth_reml pob_orig_REG pobreza com_str
sort com_id
save "$path1\pobrezas_CASEN.dta", replace
restore

preserve
replace poor_cal=sin(poor_cal)^2
replace y=sin(y)^2
replace limit_1=sin(limit_1)^2
replace limit_2=sin(limit_2)^2
replace synth_reml =sin(synth_reml)^2
replace theta_reml=sin(theta_reml)^2

keep region com_id comuna size_samp hhsample_r deff_v d y b_reml synth_reml1 theta_reml limit_1 limit_2 marca
poor_cal pobreza cal
sort com_id
save "$path1\resultados", replace
restore

* 7. Medida de incertidumbre PARTHA
*************************************************************************
preserve

matrix drop cte
matrix drop X
matrix drop w_a_reml
matrix drop aux1
matrix drop b_reml
drop aux1 b_reml
scalar drop m
scalar drop p


drop if orig2==.

mkmat y $vardep1 d
scalar m = rowsof(y)
matrix cte=J(m,1,1)
matrix X=cte,$vardep2
scalar p=colsof(X)

gen  b_reml=(d/(a_reml+d))
```

```
mkmat b_reml

gen aux1=1/(a_reml+d)
mkmat aux1
matrix w_a_reml=diag(aux1)

gen g1_i=d*(1-b_reml)

matrix aux2=inv(X'*w_a_reml*X)
matrix t2_reml_mat=X*aux2*X'

matrix aux3=J(m,1,0)

local i = 1
while `i' <=m {
matrix aux3[`i',1]=t2_reml_mat[`i',`i']
local i = `i' + 1
}

svmat aux3
gen g2_i=b_reml^2*aux3


egen part_g3_i=sum((a_reml+d)^(-2))
gen g3_i=((2 * d^2)/(a_reml+d)^3)*(part_g3_i^(-1))


gen mse_theta_reml=(g1_i+g2_i+2*g3_i)
gen rc_g1 = (g1_i/mse_theta_reml)*100
gen rc_g2 = (g2_i/mse_theta_reml)*100
gen rc_g3 = (g3_i/mse_theta_reml)*100

gen se_theta_reml = (mse_theta_reml)^(1/2)

sum rc_g1 rc_g2 rc_g3
sum g1 d

* IC
gen theta_lci_reml =  poor_cal - 1.96 * se_theta_reml
gen theta_uci_reml = poor_cal + 1.96 * se_theta_reml


sum com_id theta_lci_reml poor_cal theta_uci_reml mse_theta_reml d

sort com_id


keep com_id theta_lci_reml poor_cal theta_uci_reml mse_theta_reml se_theta_reml d rc_g1 rc_g2 rc_g3
ren com_id comuna
ren theta_lci_reml li_t_reml
ren poor_cal media_t_reml
```

```
ren  theta_uci_reml ls_t_reml
ren mse_theta_reml mse_t_reml
sort comuna
save "$path1\resultados_vt_ICreml.dta", replace
restore
```

9. Bootstrap

```
clear all
set mem 1000m
set more off
set matsize 4500
global path1 "especificar ruta de archivo donde se ubican los insumos"
* definition of numer of reps
global reps 4000
* definition of X's variables used in the weighted regression
global vardep1  f2 i1 x9 c4 x5 x14 k5 x8 c1 x4
global vardep2  f2, i1, x9, c4, x5, x14, k5, x8, c1, x4
matrix AS=J($reps,1,.)
*****************************************************************************************************
* 1.        Compute A, B and Xb from the Fay-Herriot model
*                   original theta=poor_cal
*****************************************************************************************************

* Database X is merged with the original result (theta, A, D and B)
use "$path1\BASE.dta", clear
sort com_id
merge com_id using "$path1\theta fu8ll sample.dta"
tab _m
ren _m vega


sort com_id
gen a_sd=a_reml^0.5                                                     /*varianza
asociada al modelo (A)*/
gen d_sd=d^0.5                                                          /*varianza
asociada a cada comuna (d)*/
drop  b_reml a_reml
gen y_org=y
save "$path1\BASE_boot.dta", replace

*****************************************************************************************************
* 2.        Generate bootstrap samples (N=2,000) of theta's_star and y's_star, using A, D and B from the
*                   original Fay-Herriot estimation
*****************************************************************************************************
* seed for replicate result
set seed 3246
* loop begins (N=1000)
global i "1"
forvalues i=1(1)$reps {
```

```
clear
use "$path1\BASE_boot.dta"

* Generate bootstrap samples of theta's_star
generate theta`i' = invnormal(uniform())*a_sd+poor_cal

* Generate bootstrap samples of y´s_star
generate z`i' = invnormal(uniform())*d_sd+theta`i'

        replace z`i'=0 if z`i'<0
*gen y`i'=asin(sqrt(z`i'))

drop y
gen y=z`i'
gen y`i'=y

        preserve
        keep  com_id y`i' poor_cal theta`i' d
        sort com_id
        save "$path1\boot_aux.dta", replace
        restore

keep if vega==3
saveold "$path1\base comunal v2.dta", replace

mkmat y $vardep1 d

scalar m = rowsof(y)

matrix cte=J(m,1,1)
matrix X=cte,$vardep2

scalar p=colsof(X)
```

****

* 3. Find A_star, B_star and b_star (betas) by replacing original data y´s with bootstrap
* samples y's_star (generated in step 2)

```
* RUN R FROM STATA
* solution of A using REML method
matrix A_REML=J(1,1,.)
preserve
set more off
global path1 "especificar ruta de archivo donde se ubican los insumos"
log using "$path1\A_remlprueba.log", replace
global Rterm_path "especificar ruta de archivo donde se ubica R studio"
global Rterm_options `"--vanilla"'
rsource using "$path1\ESTIMACION A_REML EN R", lsource

log close
```

```
clear

*insheet using "$path1\A_reml.log"
insheet using "$path1\A_remlprueba.log"

split v1, generate(e) parse(])

gen a_reml=real(e2)
sum a_reml

matrix A_REML[1,1]=r(mean)

restore

matrix list A_REML
di A_REML[1,1]

matrix VAR_A`i'=A_REML[1,1]
svmat VAR_A`i'
egen a_reml`i'=max(VAR_A`i')

matrix AS[`i',1]=A_REML[1,1]

gen  b_reml`i'=(d/(a_reml`i'+d))
mkmat b_reml`i'

*****************************************************************************************************
* 4.  EB estimate of theta using A_star and B_star, and bootstrap samples y's_star
*****************************************************************************************************

gen aux1=1/(a_reml`i'+d)
mkmat aux1
matrix w_a_reml=diag(aux1)

* compute betas_star
matrix beta_hat_reml=inv(X'*w_a_reml*X)*X'*w_a_reml*y

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
matrix drop X y cte d
scalar drop p m
matrix drop b_reml`i' w_a_reml
drop b_reml`i' aux1
drop y`i' theta`i' poor_cal

append using "$path1\BASE_aux.dta"
drop p*
drop m*

sort com_id
merge com_id using "$path1\boot_aux.dta"
tab _m
```

```
mkmat y`i' $vardep1 d
scalar m = rowsof(y`i')
matrix cte=J(m,1,1)
matrix X=cte,$vardep2
scalar p=colsof(X)

ren a_reml`i' a_reml`i'_aux
egen a_reml`i'=max(a_reml`i'_aux)
gen  b_reml`i'=(d/(a_reml`i'+d))
mkmat b_reml`i'

gen aux1=1/(a_reml`i'+d)
mkmat aux1
matrix w_a_reml=diag(aux1)
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*

* compute y_synth
matrix synth_reml=X*beta_hat_reml

svmat b_reml`i'
svmat synth_reml

* compute new theta
gen theta_reml`i'=(1-b_reml`i')*y`i'+b_reml`i'*synth_reml


gen p_predicho`i'=(theta_reml`i')
gen p_pred`i'=p_predicho`i'


*sum y`i' p_pred`i' p_predicho`i' theta_reml`i' synth_reml b_reml`i' d a_reml`i'

gen poor_cal`i'=p_pred`i'

*****************************************************************************************************
* 4.        For each bootstrap sample, compute t
*****************************************************************************************************

gen B_`i'=b_reml`i'
gen dsd_`i'=d^0.5
gen F_`i'=(1-B_`i')^0.5
gen t_`i'=(theta`i'-poor_cal`i')/(dsd_`i'*F_`i')
gen j_`i'=(theta`i'-poor_cal`i')^2
            *sum t_`i'
*sum theta`i' poor_cal`i' dsd_`i' F_`i' b_reml`i' a_reml`i'
            * stored t in a database
preserve
keep com_id t_`i'
sort com_id
save "$path1\T_`i'.dta", replace
```

```
		restore

		* stored j in a database
		preserve
		keep com_id j_`i'
		sort com_id
		save "$path1\J_`i'.dta", replace
		restore
		drop  theta`i'- t_`i' j_`i'
		matrix drop synth_reml beta_hat_reml w_a_reml aux1 b_reml`i' VAR_A`i' A_REML  X cte
		matrix drop $vardep1
		compress
di " "
di " "
di " "
di " "
di " "
di " "
di " "
di " "
di " "
di in red "************************************************"
di in red "  ITERACION " `i' "    " `i' "    " `i' "    "
di in red "************************************************"
di " "
di " "
di " "
di " "
di " "
di " "
di " "
di " "
di in red "************************************************"
di in red "  ITERACION " `i' "    " `i' "    " `i' "    "
di in red "************************************************"
di " "
di " "
di " "
di " "
di " "
di " "
di " "
di " "
di in red "************************************************"
di in red "  ITERACION " `i' "    " `i' "    " `i' "    "
di in red "************************************************"
di " "
di " "
di " "
di " "
```

```
di " "
di " "
di " "
di " "
di " "
}
* end of loop
*****************************************************************************************************
* 5. confidence intervals are constructed
*****************************************************************************************************

clear
set more off
* merge of t´s database
use "$path1\T_1.dta"
mkmat  t_1
drop t_1
sort com_id
forvalues x=2(1)$reps{
merge com_id using "$path1\T_`x'.dta"
tab _m
drop _m
sort com_id
mkmat t_`x'
drop t_`x'
}
* Database t´s is merged with the original result (theta, A, D and B), from Stata file named "original estimation"
merge com_id using "$path1\theta fu8ll sample.dta"
sort com_id
tab _m
drop _m
* F=(1-B)^0.5
mkmat F
* dsd=D^0.5
mkmat dsd
sort com_id
gen idcn=_n
matrix Ts=J(350,3,.)                                                    /*matriz que guarda t's
por comuna*/
set matsize 5000
forvalues m=1(1)350 {                                                   /*loop para estimar
distribución t por comuna*/
        matrix com`m'=J($reps,1,.)
        forvalues i=1(1)$reps {                                         /*transforma los datos
de las N reps=columnas, a una sola matriz por comuna*/
        matrix com`m'[`i',1]=t_`i'[`m',1]
        }
        svmat com`m'
        /*
        sum com`m', d
        matrix Ts[`m',1]=r(p5)
```

```
		matrix Ts[`m',2]=r(p95)
		*/
		egen v1_`m'=pctile(com`m'), p(2.5)
		sum v1_`m'
		matrix Ts[`m',1]=r(mean)
		egen v2_`m'=pctile(com`m'), p(97.5)
		sum v2_`m'
		matrix Ts[`m',2]=r(mean)
		drop v1_`m' v2_`m'
		sum poor_cal if idcn==`m'
		matrix Ts[`m',3]=r(mean)
		*drop com`m'
}
matrix colnames Ts = p5_boot p95_boot theta_fullsample
matrix list Ts
mkmat com_id
drop  com11- com3501
*****************************************************
* IC=theta_original+-t*(D*(1-B))^0.5
* Where thetha_original, D and B come from the original estimation,
* and t comes from the bootstrap procedure
matrix IC_pob=J(350,4,.)
forvalues m=1(1)350 {
matrix IC_pob[`m',1]=Ts[`m',3]+Ts[`m',1]*dsd[`m',1]*F[`m',1]
matrix IC_pob[`m',2]=Ts[`m',3]
matrix IC_pob[`m',3]=Ts[`m',3]+Ts[`m',2]*dsd[`m',1]*F[`m',1]
matrix IC_pob[`m',4]=com_id[`m',1]
}
matrix colnames IC_pob = IC_I MEDIA_fullsample IC_S
matrix list IC_pob
*****************************************************

preserve
svmat IC_pob
ren  IC_pob1 li_vt
ren  IC_pob2 media_vt
ren  IC_pob3 ls_vt
ren  IC_pob4 comuna
ren b_reml b_reml_vt
keep li_vt media_vt ls_vt comuna b_reml_vt size_samp size_pop* region
drop if comuna==.
sort comuna
save "$path1\resultados_vt.dta", replace
restore
matrix list AS
svmat AS
*****************************************************************************************************
* MSE
*****************************************************************************************************
clear
set more off
```

```
* merge of j´s database
use "$path1\J_1.dta"
mkmat  j_1
drop j_1
sort com_id
forvalues x=2(1)$reps{
merge com_id using "$path1\J_`x'.dta"
tab _m
drop _m
sort com_id
mkmat j_`x'
drop j_`x'
}
sort com_id
gen idcn=_n
matrix Js=J(350,1,.)                                    /*matriz que guarda t's por
comuna*/
forvalues m=1(1)350 {                                   /*loop para estimar
distribución t por comuna*/
        matrix jom`m'=J($reps,1,.)
        forvalues i=1(1)$reps {                         /*transforma los datos
de las N reps=columnas, a una sola matriz por comuna*/
        matrix jom`m'[`i',1]=j_`i'[`m',1]
        }
        svmat jom`m'
        egen s_jom`m'=sum(jom`m')
        gen MSE_`m'=s_jom`m'/$reps
        sum MSE_`m' if idcn==`m'
        matrix Js[`m',1]=r(mean)
        drop jom`m' s_jom`m' MSE_`m'
}
svmat Js
preserve
ren  com_id comuna
ren Js MSE
keep comuna MSE
drop if comuna==.
sort comuna
save "$path1\MSE_vt.dta", replace
restore
```

Grupos de pertenencia de comunas no Casen

Código	Comuna	Grupo de pertenencia
1403	Colchane	6
2202	Ollague	5
5104	Juan Fernández	26
5201	Isla de Pascua	28
10103	Cochamó	18
10401	Chaitén	17
10402	Futaleufú	16
10403	Hualaihué	18
10404	Palena	15
11102	Lago Verde	19
11203	Guaitecas	18
11302	O'Higgins	19
11303	Tortel	27
12102	Laguna Blanca	1
12103	Río Verde	22
12104	San Gregorio	2
12201	Cabo de Hornos	13
12302	Primavera	26
12303	Timaukel	23
12402	Torres del Paine	28
15202	General Lagos	6

Grupos creados a partir de un análisis de componentes principales usando variables del censo 2002.

Fuente: Observatorio Social, Ministerio de Desarrollo Social.

Grupos asignados a las comunas Casen

Código	Comuna	Grupo
1101	Iquique	11
1107	Alto Hospicio	11
1401	Pozo Almonte	22
1402	Camiña	25
1403	Colchane	6
1404	Huara	16
1405	Pica	28
2101	Antofagasta	13
2102	Mejillones	20
2103	Sierra Gorda	11
2104	Taltal	1
2201	Calama	11
2202	Ollague	5
2203	San Pedro	25
2301	Tocopilla	22
2302	María Elena	20
3101	Copiapó	2
3102	Caldera	22
3103	Tierra Amarilla	17
3201	Chañaral	1
3202	Diego de Almagro	26
3301	Vallenar	23
3302	Alto del Carmen	8
3303	Freirina	27
3304	Huasco	23
4101	La Serena	13
4102	Coquimbo	2
4103	Andacollo	27
4104	La Higuera	8
4105	Paiguano	1
4106	Vicuña	15
4201	Illapel	16
4202	Canela	12
4203	Los Vilos	5
4204	Salamanca	5
4301	Ovalle	7
4302	Combarbalá	25
4303	Monte Patria	3
4304	Punitaqui	25
4305	Río Hurtado	8
5101	Valparaíso	11
5102	Casablanca	7
5103	Concón	28
5104	Juan Fernández	26
5105	Puchuncaví	22

5107	Quintero	24
5109	Viña del Mar	21
5201	Isla de Pascua	28
5301	Los Andes	2
5302	Calle Larga	15
5303	Rinconada	15
5304	San Esteban	15
5401	La Ligua	7
5402	Cabildo	16
5403	Papudo	1
5404	Petorca	27
5405	Zapallar	1
5501	Quillón	27
5502	Calera	22
5503	Hijuelas	17
5504	La Cruz	22
5506	Nogales	15
5601	San Antonio	22
5602	Algarrobo	24
5603	Cartagena	1
5604	El Quisco	20
5605	El Tabo	20
5606	Santo Domingo	23
5701	San Felipe	20
5702	Catemu	16
5703	Llaillay	15
5704	Panquehue	17
5705	Putaendo	16
5706	Santa María	17
5801	Quilpué	13
5802	Limache	23
5803	Olmué	1
5804	Villa Alegre	17
6101	Rancagua	2
6102	Codegua	17
6103	Coinco	17
6104	Coltauco	18
6105	Doñigue	1
6106	Graneros	7
6107	Las Cabras	18
6108	Machalí	24
6109	Malloa	17
6110	Mostazal	7
6111	Olivar	17
6112	Peumo	17
6113	Pichidegua	18
6114	Quinta de Tilcoco	17
6115	Rengo	1
6116	Requínoa	17

6117	San Vicente	15
6201	Pichilemu	5
6202	La Estrella	18
6203	Litueche	25
6204	Marchihue	16
6205	Navidad	25
6206	Paredones	12
6301	San Fernando	20
6302	Chépica	3
6303	Chimbarongo	18
6304	Lolol	12
6305	Nancagua	17
6306	Palmilla	18
6307	Peralillo	27
6308	Placilla	18
6309	Pumanque	6
6310	Santa Cruz	5
7101	Talca	2
7102	Constitución	7
7103	Curepto	25
7104	Empedrado	6
7105	Maule	18
7106	Pelarco	6
7107	Pencahue	6
7108	Río Claro	6
7109	San Clemente	3
7110	San Rafael	25
7201	Cauquenes	16
7202	Chanco	3
7203	Pelluhue	25
7301	Curicó	23
7302	Hualañe	25
7303	Licantén	16
7304	Molina	16
7305	Rauco	25
7306	Romeral	27
7307	Sagrada Familia	3
7308	Teno	25
7309	Vichuquén	27
7401	Linares	22
7402	Colbún	18
7403	Longaví	3
7404	Parral	16
7405	Retiro	3
7406	San Javier	16
7407	Villa Alemana	11
7408	Yerbas Buenas	3
8101	Concepción	21
8102	Coronel	22

8103	Chiguayante	11
8104	Florida	8
8105	Hualqui	15
8106	Lota	1
8107	Penco	20
8108	San Pedro de Atacama	24
8109	Santa Juana	8
8110	Talcahuano	11
8111	Tomé	22
8112	hualpén	11
8201	Lebu	17
8202	Arauco	15
8203	Cañete	16
8204	Contulmo	25
8205	Curanilahue	16
8206	Los Alamos	3
8207	Tirúa	25
8301	Los Angeles	23
8302	Antuco	18
8303	Cabrero	18
8304	Laja	5
8305	Mulchén	18
8306	Nacimiento	16
8307	Negrete	25
8308	Quilaco	18
8309	Quilleco	3
8310	San Rosendo	17
8311	Santa Bárbara	8
8312	Tucapel	16
8313	Yumbel	27
8314	alto bio	8
8401	Chillán	2
8402	Bulnes	15
8403	Cobquecura	3
8404	Coelemu	27
8405	Coihueco	3
8406	Chillán Viejo	23
8407	El Carmen	3
8408	Ninhue	6
8409	Ñiquén	3
8410	Pemuco	3
8411	Pinto	27
8412	Portezuelo	3
8413	Quillota	24
8414	Quirihue	16
8415	Ránquil	18
8416	San Carlos	5
8417	San Fabián	12
8418	San Ignacio	18

8419	San Nicolás	6
8420	Treguaco	6
8421	Yungay	15
9101	Temuco	28
9102	Carahue	25
9103	Cunco	18
9104	Curarrehue	25
9105	Freire	18
9106	Galvarino	12
9107	Gorbea	17
9108	Lautaro	5
9109	Loncoche	16
9110	Melipeuco	3
9111	Nueva Imperial	27
9112	Padre las Casas	5
9113	Perquenco	25
9114	Pitrufquén	5
9115	Pucón	23
9116	Saavedra	6
9117	Teodoro Schmidt	25
9118	Toltén	18
9119	Vilcún	27
9120	Villarrica	7
9121	Cholchol	27
9201	Angol	5
9202	Collipulli	25
9203	Curacautín	16
9204	Ercilla	6
9205	Lonquimay	8
9206	Los Sauces	6
9207	Lumaco	12
9208	Purén	25
9209	Renaico	27
9210	Traiguen	27
9211	Victoria	5
10101	Puerto Montt	24
10102	Calbuco	18
10103	Cochamó	18
10104	Fresia	18
10105	Frutillar	15
10106	Los Muermos	3
10107	Llanquihue	7
10108	Maullín	10
10109	Puerto Varas	24
10201	Castro	22
10202	Ancud	15
10203	Chonchi	10
10204	Curaco de Vélez	10
10205	Dalcahue	10

10206	Puqueldón	10
10207	Queilén	10
10208	Quellón	17
10209	Quemchi	10
10210	Quinchao	17
10301	Osorno	24
10302	Puerto Octay	17
10303	Purranque	16
10304	Puyehue	27
10305	Río Negro	18
10306	San Juan de La Costa	6
10307	San Pablo	18
10401	Chaitén	17
10402	Futaleufú	16
10403	Hualaihué	18
10404	Palena	15
11101	Coihaique	24
11102	Lago Verde	19
11201	Aisén	7
11202	Cisnes	7
11203	Guaitecas	18
11301	Cochrane	19
11302	O'Higgins	19
11303	Tortel	27
11401	Chile Chico	7
11402	Río Ibáñez	8
12101	Punta Arenas	11
12102	Laguna Blanca	1
12103	Río Verde	22
12104	San Gregorio	2
12201	Cabo de Hornos	13
12301	Porvenir	1
12302	Primavera	26
12303	Timaukel	23
12401	Natales	1
12402	Torres del Paine	28
13101	Santiago	9
13102	Cerrillos	26
13103	Cerro Navia	1
13104	Conchalí	20
13105	El Bosque	20
13106	Estación Central	11
13107	Huechuraba	24
13108	Independencia	13
13109	La Cisterna	28
13110	La Florida	28
13111	La Granja	22
13112	La Pintana	17
13113	La Reina	4

13114	Las Condes	14
13115	Lo Barnechea	9
13116	Lo Espejo	1
13117	Lo Prado	20
13118	Macul	21
13119	Maipú	13
13120	Ñuñoa	4
13121	Pedro Aguirre Cerda	20
13122	Peñalolén	2
13123	Providencia	14
13124	Pudahuel	20
13125	Quilicura	26
13126	Quinta Normal	26
13127	Recoleta	24
13128	Renca	1
13129	San Joaquín	24
13130	San Miguel	21
13131	San Ramón	1
13132	Vitacura	14
13201	Puente Alto	26
13202	Pirque	2
13203	San José de Maipo	2
13301	Colina	7
13302	Lampa	7
13303	Tiltil	15
13401	San Bernardo	20
13402	Buin	22
13403	Calera de Tango	2
13404	Paine	1
13501	Melipilla	7
13502	Alhué	27
13503	Curacaví	22
13504	María Pinto	16
13505	San Pedro de la Paz	13
13601	Talagante	24
13602	El Monte	7
13603	Isla de Maipo	7
13604	Padre Hurtado	22
13605	Peñaflor	24
14101	Valdivia	11
14102	Corral	18
14103	Lanco	27
14104	Los Lagos	27
14105	Máfil	27
14106	Mariquina	18
14107	Paillaco	27
14108	Panguipulli	25
14201	La Unión	15
14202	Futrono	27

14203	Lago Ranco	3
14204	Río Bueno	16
15101	Arica	11
15102	Camarones	19
15201	Putre	19
15202	General Lagos	6