Muestreo 1 - Prueba 6

Daniel Czarnievicz

Julio 2017

A continuación se presentan los resultados de la prueba 6. La respuesta 3.1 aparece dos veces dado que así figuraba en la pauta de la prueba.

Cargamos la base de datos del archivo `U.txt`, la ordenamos, y definimos la variable `N` (el tamaño poblacional).

```
U <- read.table("U.txt", header=T)
U <- arrange(U, id)
N <- dim(U)[1]
```

PARTE 1: diseño SI

Se carga la muestra seleccionada (lista de índices), y se seleccionan las observaciones correspondientes. Se genera el valor `n`, correspondiente al tamaño muestral.

```
ssi <- read.table("sSI.txt", header=T)
datossi <- U[ssi$id,]
n <- dim(datossi)[1]
```

Se generan las variables `fpc` (tamaño de la población), `pw`, y `pik` necesarias para los cálculos.

```
datossi$fpc <- rep(N, n)
datossi$pw <- rep(N/n, n)
datossi$pik <- rep(n/N, n)
```

1.1

Se estiman el estimador $\hat{t}_\pi$ y su varianza $\hat{V}(\hat{t}_\pi)$.

```
svytotal(~y, svydesign(id=~1, data=datossi, fpc=~fpc))
```

```
   total    SE
y 178327 20131
```

1.2

La estimación de $\hat{V}_0$ puede hacerse de dos manera:

1. indicando los pesos $\frac{1}{\pi_k}$ (contenidos en la vaiable `pw`), o
2. indicando las probabilidades de inclusión π_k (contenidas en la variable `pik`).

```
svytotal(~y, svydesign(id=~1, data=datossi, weights=~pw))
```

```
   total    SE
y 178327 21121
```

```
svytotal(~y, svydesign(id=~1, data=datossi, probs=~pik))
```

```
   total    SE
y 178327 21121
```

Para la parte 1 no se utilizó información auxiliar.

PARTE 2: diseño SY

Primero se debe ordenar la población en función de la variable x. Se genera un nuevo id, idx, con la población ordenada de menor a mayor según los valores de x.

```
Ux <- arrange(U, x)
Ux$idx <- seq(1, dim(Ux)[1], 1)
```

Se carga la muestra seleccionada (lista de índices), y se seleccionan las observaciones correspondientes. Se genera el valor n, correspondiente al tamaño muestral.

```
ssy <- read.table("sSY.txt", header=T)
datossy <- Ux[ssy$id,]
n <- dim(datossy)[1]
```

2.1

Para determinar el intervalo de muestreo se calcula el salto entre un índice y el siguiente. El arranque aleatorio fue simplemente el primer índice seleccionado en la base ordenada según x. Esto se corresponde con la observación 315 en la población original.

```
a <- vector("numeric", n-1)
for(i in 1:n-1){
      a[i] <- datossy$idx[i+1] - datossy$idx[i]
}
(a <- unique(a)) # Intervalo de muestreo
```

```
[1] 11
```

```
ssy[1,1] # Arranque aleatorio
```

```
[1] 4
```

```
Ux[Ux$idx==ssy[1,1],"id"]
```

```
[1] 315
```

2.2

Se estiman el estimador $\hat{t}_\pi$ y su desvío $\sqrt{\hat{V}_{SI}(\hat{t}_\pi)}$, asumiendo un diseño SI.

```
# Estmador de t_y
a*sum(datossy$y)
```

```
[1] 165132
```

```
# Estimador del desvío de t_y
sqrt((N^2/n)*(1-n/N)*var(datossy$y))
```

```
[1] 19264.95
```

2.3

La estimación de $\hat{V}_0$ puede hacerse de dos manera:

1. indicando los pesos $\frac{1}{\pi_k} = a$ (contenidos en la vaiable pw), o
2. indicando las probabilidades de inclusión $\pi_k = \frac{1}{a}$ (contenidas en la variable pik).

```
datossy$pw <- rep(a, n)
svytotal(~y, svydesign(id=~1, data=datossy, weights=~pw))
```

```
   total    SE
y 165132 20356
```

```
datossy$pik <- rep(1/a, n)
svytotal(~y, svydesign(id=~1, data=datossy, probs=~pik))
```

```
   total    SE
y 165132 20356
```

Se utilizió como información auxiliar para el diseño a la variable x.

PARTE 3: diseño πps

Se carga la muestra seleccionada (lista de índices), y se seleccionan las observaciones correspondientes. Se genera el valor n, correspondiente al tamaño muestral.

```
spips <- read.table("sPIPS.txt", header=T)
datospips <- U[spips$id,]
n <- dim(datospips)[1]
```

3.1

Para realizar la estimación usando las funciones de la librería `survey`, primero se calculan los π_k.

```
datospips$pik <- vector("numeric", n)
for(i in 1:n){
      datospips$pik[i] <- n * (datospips$x[i] / sum(U$x))
}
svytotal(~y, svydesign(id=~1, data=datospips, probs=~pik))
```

```
   total     SE
y 141970 6061.3
```

PARTE 3: diseño STSI

Se carga la muestra seleccionada (lista de índices), y se seleccionan las observaciones correspondientes. Se genera el valor n, correspondiente al tamaño muestral. Se le agregan los estrados a cada observación (variable $ST1$).

```
sstsi <- read.table("sST1.txt", header=T)
datosstsi <- U[sstsi$id,]
datosstsi$ST1 <- sstsi$ST1
n <- dim(datosstsi)[1]
```

3.1

Para determinar el criterio utilizado para asignar los tamaños de muestra por estrado se calculan cuáles deberían haber sido los tamaños (n_h) si los criterios hubieran sido proporcional y óptimo, respectivamente:

```
# Cantidad de observaciones por estrato (criterio: proporcional)
group_by(datosstsi, ST1) %>% summarise(obs=n(), varx=var(x), vary=var(y))
```

```
# A tibble: 4 x 4
    ST1   obs    varx   vary
  <int> <int>   <dbl>  <dbl>
1     1    22    74.0   224.
2     2    22   136.    987.
3     3    23   673.   5817.
```

```
4     4    23 13229.  41311.
```

```
90/4
```

```
[1] 22.5
```

```
# Corroborando que no sea proporcional a la varianza por estrato de x
U$quantx <- vector("numeric", N)
for(i in 1:N){
      if(quantile(U$x)[1] <= U$x[i] & U$x[i] <= quantile(U$x)[2]){
            U$quantx[i] <- 1
      }else if(quantile(U$x)[2] < U$x[i] & U$x[i] <= quantile(U$x)[3]){
            U$quantx[i] <- 2
      }else if(quantile(U$x)[3] < U$x[i] & U$x[i] <= quantile(U$x)[4]){
            U$quantx[i] <- 3
      }else{U$quantx[i] <- 4}
}
grupos <- as.data.frame(group_by(U, quantx) %>% summarise(N_h=n(), varx=var(x), nvar=sum(n()*var(x))))
sumx = as.numeric(grupos[4,"nvar"])
group_by(U, quantx) %>% summarise(N=n(), varx=var(x), nx=round((N*n()*var(x))/sumx,2))
```

```
# A tibble: 4 x 4
  quantx     N   varx     nx
   <dbl> <int>  <dbl>  <dbl>
1      1   246   71.0   2.01
2      2   246  132.    3.75
3      3   245  529.   14.9
4      4   246 8666.  246
```

Por lo tanto, el criterio utilizado fue el de tamaños muestrales proporcionales al tamaño del estrado. En este citerio $n_h = n\,\frac{N_h}{N}$. Dado que los estratos se construyeron según los cuartiles de la variable x, la misma constituye información auxiliar utilizada para el muestreo.

3.2

Se estiman el estimador $\hat{t}_\pi$ y su desvío $\sqrt{\hat{V}_{STSI}(\hat{t}_\pi)}$.

```
datosstsi <- left_join(datosstsi, grupos[c("quantx","N_h")], by=c("ST1" = "quantx"))
svytotal(~y, svydesign(id=~1, strata=~ST1, data=datosstsi, fpc=~N_h))
```

```
   total    SE
y 153294 10739
```

3.3

Estimación de $\hat{V}_0$.

```
n_h <- as.data.frame(group_by(datosstsi, ST1) %>% summarise(n_h=n()))
datosstsi <- left_join(datosstsi, n_h, by="ST1")
datosstsi$pik <- datosstsi$n_h / datosstsi$N_h
svytotal(~y, svydesign(id=~1, data=datosstsi, strata=~ST1, probs=~pik))
```

```
   total    SE
y 153294 11279
```

PARTE 4: diseño STSI

4.1

Para tomar la muestra primero se ordena la población, U, en función de la variable x. Luego se utilizó la función `strata.cumrootf` de la librería `stratification`, la cual implementa el criterio de la $\sqrt{f}$ de Dalenius para la construcción de los estratos.

```
U <- U[order(U$x),]
boundaries <- strata.cumrootf(U$x, n=90, Ls=6)
U$h6 <- boundaries$stratumID
Nh <- boundaries$Nh
nh <- boundaries$nh
```

Para tomar la muestra, primero se fija una semilla. De esta forma los resultados son reproducibles.

```
set.seed(100)
ids <- sampling::strata(data=U, stratanames="h6", size=nh, method="srswor")
muestra <- U[ids$ID_unit,]
```

4.2

Para estimar $\hat{t}_\pi$ y su desvío, la función `svytotal` requiere de los tamaños poblcionales (o las tasas de muestreo) por estrato, N_h. En el siguiente código se construyen vectores con esta información y con los tamaños muestrales por estrato, n_h (los cuales se utilizaron en **4.3**). Ambos vectores se agregan a la muestra.

```
n_h <- rep(nh[1], nh[1])
for(i in 2:length(nh)){
      n_h <- c(n_h, rep(nh[i], nh[i]))
}

N_h <- rep(Nh[1], nh[1])
for(i in 2:length(Nh)){
      N_h <- c(N_h, rep(Nh[i], nh[i]))
}
muestra <- cbind(muestra, n_h, N_h)
svytotal(~y, svydesign(id=~1, strata=~h6, data=muestra, fpc=~N_h))
```

```
   total     SE
y 150290 6023.3
```

4.3

Para estimar el estimador $\hat{V}_0$ primero se calculan los π_{k_h} para cada estrato h.

```
muestra$pik <- n_h/N_h
svytotal(~y, svydesign(id=~1, strata=~h6, data=muestra, probs=~pik))
```

```
   total     SE
y 150290 6485.1
```

Al igual que en la **PARTE 3** la información de la variable x se utilizó como información auxiliar en el diseño.

PARTE 5: Estimador de razón

Estimación de $\hat{t}_{yra}$.

```
as.numeric(svyratio(~y, ~x, svydesign(id=~1, data=datossi, fpc=~fpc))$ratio)*sum(U$x)
```

```
[1] 147233.7
```

Estimación de $\hat{V}_{SI}(\hat{t}_{yra})$.

```
sqrt(as.numeric(svyratio(~y, ~x, svydesign(id=~1, data=datossi, fpc=~fpc))$var)*(sum(U$x)^2))
```

```
[1] 6006.799
```

Estimación de $\hat{V}_0(\hat{t}_{yra})$.

```
sqrt(as.numeric(svyratio(~y, ~x, svydesign(id=~1, data=datossi, weights=~pw))$var)*(sum(U$x)^2))
```

```
[1] 6302.229
```

En este caso, la información de la variable x es utilizada para la estimación.

PARTE 6: Comparación de resultados

Diseño	$\hat{t}_\pi$	$\sqrt{\hat{V}_{p(s)}(\hat{t}_\pi)}$	$\sqrt{\hat{V}_0}$	$\hat{t}_{yra}$	$\sqrt{\hat{V}_{p(s)}(\hat{t}_{yra})}$	$\sqrt{\hat{V}_0(\hat{t}_{yra})}$
SI	178 327	20 131	21 121	147 233	6 006	6 302
SY	165 132	19 264	20 356			
πps	141 973	6 061				
$STSI_{h=4}$	153 294	10 739	11 279			
$STSI_{h=6}$	138 758	6 317	6 737			

Tal como es de esperar, los diseños que utilizan la información auxiliar (variable x en este caso) obtienen mejores resultados. La estimación puede mejorarse aún más haciendo uso de la información auxiliar para la estimación. De hecho, bajo el diseño SI (diseño que no utiliza la información auxiliar), la estimación utilizando el estimador $\hat{t}_{yra}$ es comparable con la estimación de $\hat{t}_\pi$ bajo un diseño πps. Esto se debe a la alta correlación entre y y x.

```
cor(U$x, U$y)
```

```
[1] 0.9135655
```