STAT 4234/5234: Calculating the ratio estimator for a population total

Consider the population $\{(x_i, y_i) : i = 1, \ldots, N\}$ and suppose we wish to estimate the population mean $\bar{y}_U$ based on a simple random sample of size n. We further suppose the value of $\bar{x}_U$, the population mean for the auxiliary variable, is known. In ratio estimation we estimate $\bar{y}_U$ by

$$\hat{\bar{y}}_r = \hat{B}\bar{x}_U = \frac{\bar{y}}{\bar{x}}\bar{x}_U \ .$$

We have further seen that the MSE of $\hat{\bar{y}}_r$ can be estimate by

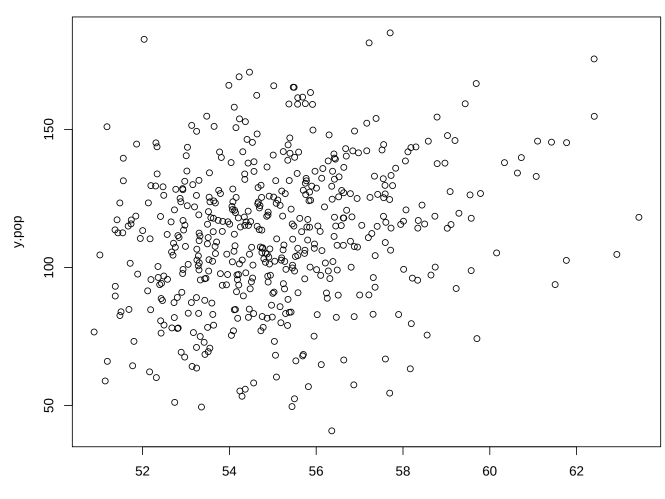
$$\hat{V}\left(\hat{\bar{y}}_r\right) = \left(\frac{\bar{x}_U}{\bar{x}}\right)^2 \frac{s_e^2}{n}\left(1 - \frac{n}{N}\right)$$

where s_e^2 is the sample variance of the $e_i = y_i - \hat{B}x_i$. The standard error of $\hat{\bar{y}}_r$ is then taken to be the square root of $\hat{V}$.

To illustrate the computing for ratio estimation, we create a fictional population of (x_i, y_i) as follows.

```
> set.seed(5234)
> x.pop <- 50 + rgamma(500, shape=5, rate=1)
> y.pop <- rnorm(500, mean=2*x.pop, sd=25)
> plot(x.pop, y.pop)
> cor(x.pop, y.pop)
[1] 0.2267459
> mean(y.pop) / mean(x.pop)
[1] 2.029948
> xbar.U <- mean(x.pop); xbar.U;
[1] 55.09557
```

The population correlation coefficient is $R = 0.23$, not particularly strong. The population ratio value is $B = \bar{y}_U/\bar{x}_U = 2.03$, and the auxiliary variable population mean is $\bar{x}_U = 55.10$.



Now we take a simple random sample of size $n = 25$.

```
> N <- 500; n <- 25;
> samp <- sample(N, n)
> x.samp <- x.pop[samp]
> y.samp <- y.pop[samp]
```

Calculate the ratio estimator $\hat{\bar{y}}_r$.

```
> xbar <- mean(x.samp); ybar <- mean(y.samp);
> xbar; ybar;
[1] 55.58985
[1] 117.2632
> B.hat <- ybar / xbar; B.hat;
[1] 2.109436
> ybar.hat.r <- B.hat * xbar.U; ybar.hat.r;
[1] 116.2206
```

And the standard error of our estimate.

```
> e <- y.samp - B.hat * x.samp
> V.hat <- (xbar.U/xbar)^2 * var(e)/n * (1 - n/N)
> SE <- sqrt(V.hat); SE;
[1] 5.559118
```

Now a 95% confidence interval for $\bar{y}_U$ is

```
> ybar.hat.r + c(-1,1) * 1.96 * SE
[1] 105.3247 127.1164
```

And a 95% confidence interval for the population total t_y is

```
> N * ( ybar.hat.r + c(-1,1) * 1.96 * SE )
[1] 52662.35 63558.22
```

Here's an R function that takes the sample data as inputs, along with N and $\bar{x}_U$, and returns the ratio estimator of $\bar{y}_U$ along with its standard error.

```
ratio.estimator.mean <- function(x.samp, y.samp, N, xbar.U)
{
 n <- length(y.samp)
 xbar <- mean(x.samp); ybar <- mean(y.samp);
 B.hat <- ybar / xbar
 ybar.hat.r <- B.hat * xbar.U
 e <- y.samp - B.hat * x.samp
 V.hat <- (xbar.U/xbar)^2 * var(e)/n * (1 - n/N)
 SE <- sqrt(V.hat)
 answer <- c(point.est=ybar.hat.r, std.error=SE)
 return(answer)
}
```

You can use this function for your homework if you wish.

```
> result <- ratio.estimator.mean(x.samp=x.samp, y.samp=y.samp,
+   N=N, xbar.U=xbar.U)
> result
 point.est  std.error
116.220565   5.559118
```

A 95% confidence interval for $\bar{y}_U$ is

```
> result[1] + c(-1,1) * 1.96 * result[2]
[1] 105.3247 127.1164
```

and a 95% CI for the population total t_y is

```
> N * ( result[1] + c(-1,1) * 1.96 * result[2] )
[1] 52662.35 63558.22
```

Domain estimation

Suppose each member of the population belongs to exactly one of D domains, and we wish to estimate $\bar{y}_{U_d}$, the population mean in domain d. A natural estimator is the domain d sample mean $\bar{y}_d$, and the MSE can be estimate by

$$\hat{V}(\bar{y}_d) = \frac{n(n_d-1)}{n_d(n-1)}\frac{s_{yd}^2}{n_d}\left(1-\frac{n}{N}\right)$$

where n_d is the (random) number of domain d subjects in the sample, and s_{yd}^2 is the domain d sample variance. The standard error is then the square root of $\hat{V}$.

To illustrate the computing, let's take our made up population above and suppose there are 3 domains, corresponding to $x \leq 54$, $54 < x \leq 56$, and $x > 56$.

```
> domain.samp <- rep(NA, n)
> for(i in 1:n)
+ {
+  if(x.samp[i] <= 54){ domain.samp[i] <- 1 }
+  else{ if(x.samp[i] <= 56){ domain.samp[i] <- 2 }
+        else{ domain.samp[i] <- 3 }
+      }
+ }
> domain.samp
 [1] 2 1 3 1 2 3 2 2 2 2 2 3 3 3 2 2 1 2 2 2 3 1 3 3 3
> table(domain.samp)
domain.samp
 1  2  3
 4 12  9
```

Suppose we are interested in estimating the domain 2 population mean, that is, the mean value of y among those cases where $54 < x \leq 56$.

```
> d <- 2
> n.d <- sum(domain.samp==d); n.d;
[1] 12
> y.samp.d <- y.samp[domain.samp==d]
> ybar.d <- mean(y.samp.d); ybar.d;
[1] 121.2192
```

Our point estimate is $\bar{y}_d = 121.22$, the average value of the $n_d = 12$ observations in our sample from domain $d = 2$.

```
> s2.yd <- var(y.samp.d)
> V.hat <- n*(n.d-1) / (n.d*(n-1)) * s2.yd/n.d * (1 - n/N)
> SE <- sqrt(V.hat); SE;
[1] 5.788222
```

And the standard error of our estimate is $\text{SE}(\bar{y}_d) = 5.79$.

Here is an R function that takes as inputs the sample data `y.samp`, as well as a `domain.samp` indicating domain membership, and returns an estimate of the domain `d` mean along with its standard error.

```
domain.estimation <- function(y.samp, domain.samp, d, N)
{
 n <- length(y.samp); n.d <- sum(domain.samp==d);
 y.samp.d <- y.samp[domain.samp==d]
 ybar.d <- mean(y.samp.d); s2.yd <- var(y.samp.d);
 V.hat <- n*(n.d-1)/(n.d*(n-1)) * s2.yd/n.d * (1 - n/N)
 SE <- sqrt(V.hat)
 answer <- c(point.est=ybar.d, std.error=SE)
 return(answer)
}
```

You can use this function for your homework if you wish.

```
> domain.estimation(y.samp=y.samp, domain.samp=domain.samp, d=d, N=N)
 point.est  std.error
121.219240   5.788222
```