survey sampling a2

Ifeakachukwu Ovili

2024-11-02

```r
# Load the SHS dataset
data("SHS", package = "stratification")

# Check the structure of the dataset
str(SHS)
```

```
## 'data.frame':    16057 obs. of  7 variables:
##  $ CASEID  : int  2395 1970 4623 6603 1441 2682 1552 3635 5693 2394 ...
##  $ WEIGHT  : int  111 98 183 253 78 122 83 153 222 111 ...
##  $ PROVINCP: int  10 10 10 10 10 10 10 10 10 10 ...
##  $ URBRUR  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ URBSIZEP: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ HHINCTOT: num  16000 30000 120000 45000 71000 100000 54000 18000 31000
13000 ...
##  $ M101    : num  744 1032 2978 694 6040 ...
```

```r
# Create Stratum column based on Province
SHS$Stratum <- SHS$PROVINCP  # Stratify based on PROVINCP
str(SHS)
```

```
## 'data.frame':    16057 obs. of  8 variables:
##  $ CASEID  : int  2395 1970 4623 6603 1441 2682 1552 3635 5693 2394 ...
##  $ WEIGHT  : int  111 98 183 253 78 122 83 153 222 111 ...
##  $ PROVINCP: int  10 10 10 10 10 10 10 10 10 10 ...
##  $ URBRUR  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ URBSIZEP: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ HHINCTOT: num  16000 30000 120000 45000 71000 100000 54000 18000 31000
13000 ...
##  $ M101    : num  744 1032 2978 694 6040 ...
##  $ Stratum : int  10 10 10 10 10 10 10 10 10 10 ...
```

```r
# Load necessary packages
library(stratification)
```

```
##
## Attaching package: 'stratification'
```

```
## The following object is masked _by_ '.GlobalEnv':
##
##     SHS
```

```
library(splitstackshape)
library(sampling)

# Create a new column for strata based on Province
SHS$Stratum <- SHS$PROVINCP

# Compute the population stratum sizes
Nh <- table(SHS$Stratum)

# Define nh (sample sizes for each stratum)
nh <- c(97, 133, 127, 150, 170, 141, 108, 150, 170, 80)

# Ensure nh is a named vector matching the strata
names(nh) <- unique(SHS$Stratum)

# Draw sample using STSRSWOR directly from SHS
set.seed(123)  # Set seed for reproducibility
Sample.stsrswor <- strata(SHS, "Stratum", size = nh, method = "srswor")

# Merge with the original SHS dataset to include M101
Sample.stsrswor <- merge(Sample.stsrswor, SHS[, c("CASEID", "M101")],
                         by.x = "ID_unit", by.y = "CASEID", all.x = TRUE)
```

```
## Warning in merge.data.frame(Sample.stsrswor, SHS[, c("CASEID", "M101")], :
## column name 'Stratum' is duplicated in the result
```

```
# Rename the duplicated 'Stratum' column
names(Sample.stsrswor)[which(duplicated(names(Sample.stsrswor)))] <-
"Stratum_ID"

# Check the sample structure
str(Sample.stsrswor)
```

```
## 'data.frame':    1326 obs. of  5 variables:
##  $ ID_unit   : int  13 26 34 41 67 69 72 90 121 141 ...
##  $ Stratum   : int  10 10 10 10 10 10 10 10 10 10 ...
##  $ Prob      : num  0.0682 0.0682 0.0682 0.0682 0.0682 ...
##  $ Stratum_ID: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ M101      : num  1800 473 2470 25155 NA ...
```

```
# Remove rows with NA in M101
Sample.stsrswor <- Sample.stsrswor[!is.na(Sample.stsrswor$M101), ]

# Final structure
str(Sample.stsrswor)
```

```
## 'data.frame':    1265 obs. of  5 variables:
##  $ ID_unit   : int  13 26 34 41 69 72 90 121 141 153 ...
##  $ Stratum   : int  10 10 10 10 10 10 10 10 10 10 ...
##  $ Prob      : num  0.0682 0.0682 0.0682 0.0682 0.0682 ...
##  $ Stratum_ID: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ M101      : num  1800 473 2470 25155 2729 ...
```

```
# Estimate the population mean Y¯U
Y_bar_U <- mean(Sample.stsrswor$M101, na.rm = TRUE)
print(paste("Estimated Mean Expenditure: ", Y_bar_U))
```

```
## [1] "Estimated Mean Expenditure:  3301.00239525692"
```

```
## variance of tyh.hat in each stratum

s2.yh=aggregate(M101 ~ Stratum_ID, Sample.stsrswor, var)[,2]
s2.yh
```

```
##  [1] 18911338 10718680 24285311 23030071 10096720 19926654 41424449
43624130
##  [9] 12037312 27717654
```

```
## variance of tyh.hat in each stratum

Nh^2*(1-nh/Nh)*s2.yh/nh
```

```
##
##           10           12           13           24           35
46
## 367338115407 174861802183 383454349480 594985727603 247708838571
283661146072
##           47           48           59           60
## 760355357844 782482825489 251351174658 199040822320
```

```
## variance of the population total estimator

sum(Nh^2*(1-nh/Nh)*s2.yh/nh)
```

```
## [1] 4.04524e+12
```

```
# Calculate the variance of the total estimator
var_total <- sum((Nh^2 * (1 - nh/Nh) * s2.yh / nh), na.rm = TRUE)

# Calculate the standard error (SE)
n <- sum(nh)  # Total sample size
standard_error <- sqrt(var_total / n)

# Determine the critical value for a 95% confidence interval
alpha <- 0.05
critical_value <- qt(1 - alpha / 2, df = sum(nh) - 1)
```

```r
# Calculate the confidence interval
lower_bound <- Y_bar_U - critical_value * standard_error
upper_bound <- Y_bar_U + critical_value * standard_error

# Print the confidence interval
confidence_interval <- c(lower_bound, upper_bound)
print(paste("95% Confidence Interval for ȲU: [", lower_bound, ", ",
upper_bound, "]"))
```

```
## [1] "95% Confidence Interval for ȲU: [ -105053.105201472 ,
111655.109991986 ]"
```