Data Analysis and Simulation for Simple Random Sampling

Longhai Li

September 2024

Contents

Analysis of agsrs.csv Data **1**
Step by step calculation without using a function . . . 1
Write a function for repeated use . . . 2
Comparing with true value . . . 3

A Simulation Demonstration of SRS Inference **3**
True Values . . . 3
One SRS sampling . . . 4
Repeating SRS sampling 5000 times . . . 4
Empirical Coverage Rate of CIs . . . 7

```r
library(latex2exp)
```

Analysis of agsrs.csv Data

Step by step calculation without using a function

```r
## read survey data
agsrs <- read.csv ("data/agsrs.csv")
head(agsrs)
```

```
##               county state acres92 acres87 acres82 farms92 farms87 farms82
## 1     COFFEE COUNTY    AL  175209  179311  194509     760     842     944
## 2    COLBERT COUNTY    AL  138135  145104  161360     488     563     686
## 3      LAMAR COUNTY    AL   56102   59861   72334     299     362     447
## 4    MARENGO COUNTY    AL  199117  220526  231207     434     471     622
## 5     MARION COUNTY    AL   89228  105586  113618     566     658     748
## 6 TUSCALOOSA COUNTY    AL   96194  120542  134616     436     521     650
##   largef92 largef87 largef82 smallf92 smallf87 smallf82 region
## 1       29       28       21       57       47       66      S
## 2       37       41       42       12       44       47      S
## 3        4        4        3       16       20       30      S
## 4       48       66       62       14       11       28      S
## 5        7        9        9       11       23       27      S
## 6       20       17       23       18       32       29      S
```

```r
## extract the variable of interest
sdata <- agsrs$acres92
N <- 3078
```

```r
## do calculation
n <- length (sdata)
ybar <- mean (sdata)
se.ybar <- sqrt((1 - n / N)) * sd (sdata) / sqrt(n)
mem <- qt (0.975, df = n - 1) * se.ybar
## return estimate vector for pop mean
c (Est. = ybar, S.E. = se.ybar, ci.low = ybar - mem, ci.upp = ybar + mem)
```

```
##      Est.      S.E.    ci.low    ci.upp
## 297897.05  18898.43 260706.26 335087.84
```

```r
## return estimate vector for pop total
c (Est. = ybar, S.E. = se.ybar, ci.low = ybar - mem, ci.upp = ybar + mem) * N
```

```
##       Est.       S.E.     ci.low     ci.upp
##  916927110   58169381  802453859 1031400361
```

Write a function for repeated use

A function for doing data analysis for srs sample

```r
#
# sdata -- a vector of sampling survey data
# N -- population size
# to find total, multiply N to the estimate returned by this function
srs_mean_est <- function (sdata, N)
{
    n <- length (sdata)
    ybar <- mean (sdata)
    se.ybar <- sqrt((1 - n / N)) * sd (sdata) / sqrt(n)
    mem <- qt (0.975, df = n - 1) * se.ybar
    c (ybar = ybar, se = se.ybar, ci.low = ybar - mem, ci.upp = ybar + mem)
}
```

Apply srs_mean_est to agsrs.csv data

Import Data

```r
agsrs <- read.csv ("data/agsrs.csv")
```

Estimating the mean of acre92

```r
srs_mean_est (agsrs[,"acres92"], N = 3078)
```

```
##      ybar        se    ci.low    ci.upp
## 297897.05  18898.43 260706.26 335087.84
```

Estimating the total of acre92

```r
srs_mean_est (agsrs[,"acres92"], N = 3078) * 3078
```

```
##       ybar         se     ci.low     ci.upp
##  916927110   58169381  802453859 1031400361
```

Estimating the proportion of counties with fewer than 200K acres for farming in 1992

```r
acres92.is.fewer.200k <- as.numeric (agsrs[,"acres92"] < 200000)
head(acres92.is.fewer.200k)
```

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

```r
srs_mean_est (acres92.is.fewer.200k, N = 3078)
```

```
##       ybar         se     ci.low     ci.upp
## 0.51000000 0.02746498 0.45595084 0.56404916
```

Estimating the total number of counties with fewer than 200K acres for farming in 1992

```r
srs_mean_est (acres92.is.fewer.200k, N = 3078) * 3078
```

```
##       ybar         se     ci.low     ci.upp
## 1569.78000   84.53722 1403.41670 1736.14330
```

Comparing with true value

```r
agpop <- read.csv ("data/agpop.csv", na = "-99")
#true mean
mean (agpop[, "acres92"], na.rm = T)
```

```
## [1] 308582.4
```

```r
# true total
sum (agpop[, "acres92"], na.rm = T)
```

```
## [1] 943953599
```

```r
# true proportion of counties with less than 200K acres for farming
mean (agpop[, "acres92"] < 200000, na.rm = T)
```

```
## [1] 0.5145472
```

```r
# true number of counties with less than 200K acres for farming
sum (agpop[, "acres92"] < 200000, na.rm = T)
```

```
## [1] 1574
```

A Simulation Demonstration of SRS Inference

```r
# read population data
agpop <- read.csv ("data/agpop.csv")
# remove those counties with na
agpop <- subset( agpop, acres92 != -99)
```

True Values

```r
# sample size
n <- 300
# population size
N <- nrow (agpop); N
```

```
## [1] 3059
```

```r
# true value of population mean
ybarU <- mean (agpop[,"acres92"]); ybarU
```

```
## [1] 308582.4
```

```r
# true value of deviation of sample mean
true.se.ybar <- sqrt (1- n/N) * sd (agpop[,"acres92"]) / sqrt (n); true.se.ybar
```

```
## [1] 23320.29
```

One SRS sampling

```r
##
# srs sampling
srs <- sample (1:N,n)
head(agpop [srs, ])
```

```
##                county state acres92 acres87 acres82 farms92 farms87 farms82
## 2797 CHARLOTTE COUNTY    VA  112944  118811  131676     451     518     686
## 2590      HALL COUNTY    TX  443027  393949  458988     297     296     341
## 1027      KNOX COUNTY    KY   46321   51153   56086     376     379     446
## 1680   LINCOLN COUNTY    NC   58384   59491   69404     425     480     560
## 166      BUTTE COUNTY    CA  452347  494530  467426    1944    2030    1785
## 55      MONROE COUNTY    AL  110066  149361  153040     400     455     540
##      largef92 largef87 largef82 smallf92 smallf87 smallf82 region
## 2797       15       19       15       31       53       72      S
## 2590      107      103      127       12       13       12      S
## 1027        4        2        2       39       32       27      S
## 1680        5        5        4       14       25       23      S
## 166        72       90      109      531      553      489      W
## 55         18       25       21       12       35       41      S
```

```r
# get data of variable "acres92"
sdata <- agpop [srs, "acres92"]
# analysis
srs_mean_est (sdata, N)
```

```
##      ybar        se    ci.low    ci.upp
## 309308.32  20652.23 268666.18 349950.47
```

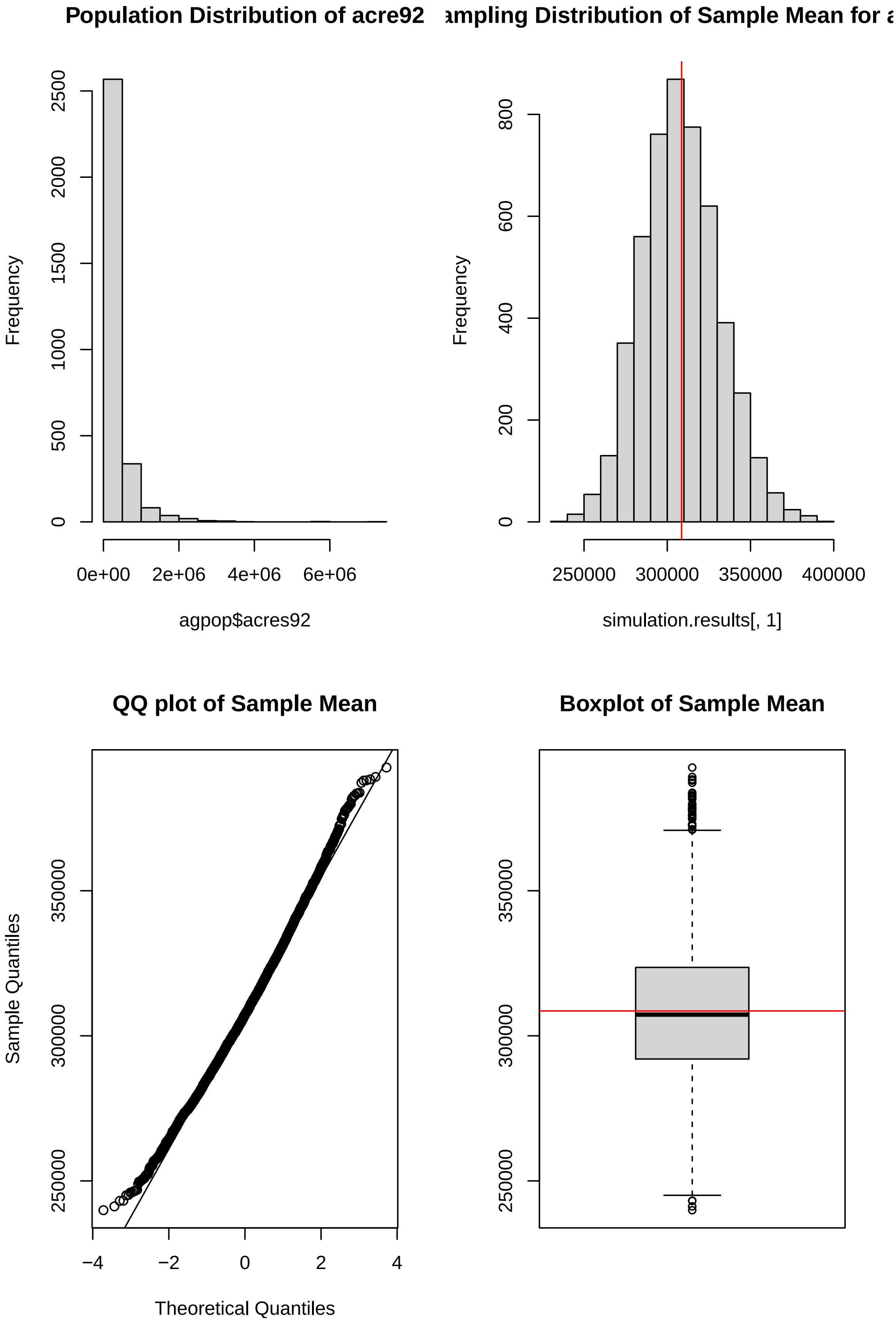
Repeating SRS sampling 5000 times

```r
nres <- 5000 # number of repeated sampling
simulation.results <- matrix (0, nres, 4) # matrix recording repeated results
colnames(simulation.results) <- c( "Est.",   "S.E.",   "ci.low", "ci.upp")

for (i in 1:nres)
{
    srs <- sample (N, n)
    sdata <- agpop [srs, "acres92"]
    simulation.results [i,] <- srs_mean_est (sdata, N)
}

head(simulation.results)
```

```
##          Est.     S.E.   ci.low   ci.upp
## [1,] 288511.1 19565.96 250006.7 327015.5
## [2,] 325070.6 22549.09 280695.6 369445.6
## [3,] 344123.0 34113.49 276990.0 411255.9
## [4,] 310826.7 21721.46 268080.4 353573.1
```

```
## [5,] 306543.3 23164.41 260957.3 352129.2
## [6,] 252130.9 15800.82 221036.0 283225.8
```

```r
# look at the distribution of sample mean
par (mfrow= c(2,2))
hist (agpop$acres92,main = "Population Distribution of acre92")
hist (simulation.results[,1], main = "Sampling Distribution of Sample Mean for acre92")
abline (v = ybarU, col = "red")
qqnorm (simulation.results[,1], main="QQ plot of Sample Mean"); qqline(simulation.results[,1])
boxplot (simulation.results[,1], main = "Boxplot of Sample Mean")
abline (h = ybarU, col = "red")
```


Population Distribution of acre92
Frequency
0
500
1000
1500
2000
2500
0e+00
2e+06
4e+06
6e+06
agpop$acres92
ampling Distribution of Sample Mean for a
Frequency
0
200
400
600
800
250000
300000
350000
400000
simulation.results[, 1]
QQ plot of Sample Mean
Sample Quantiles
250000
300000
350000
−4
−2
0
2
4
Theoretical Quantiles
Boxplot of Sample Mean
250000
300000
350000

```r
mean (simulation.results[,1])
```

```
## [1] 308270.8
```

```r
ybarU
```

```
## [1] 308582.4
```

```r
sd (simulation.results [,1])
```

```
## [1] 23367.04
```

```r
true.se.ybar
```

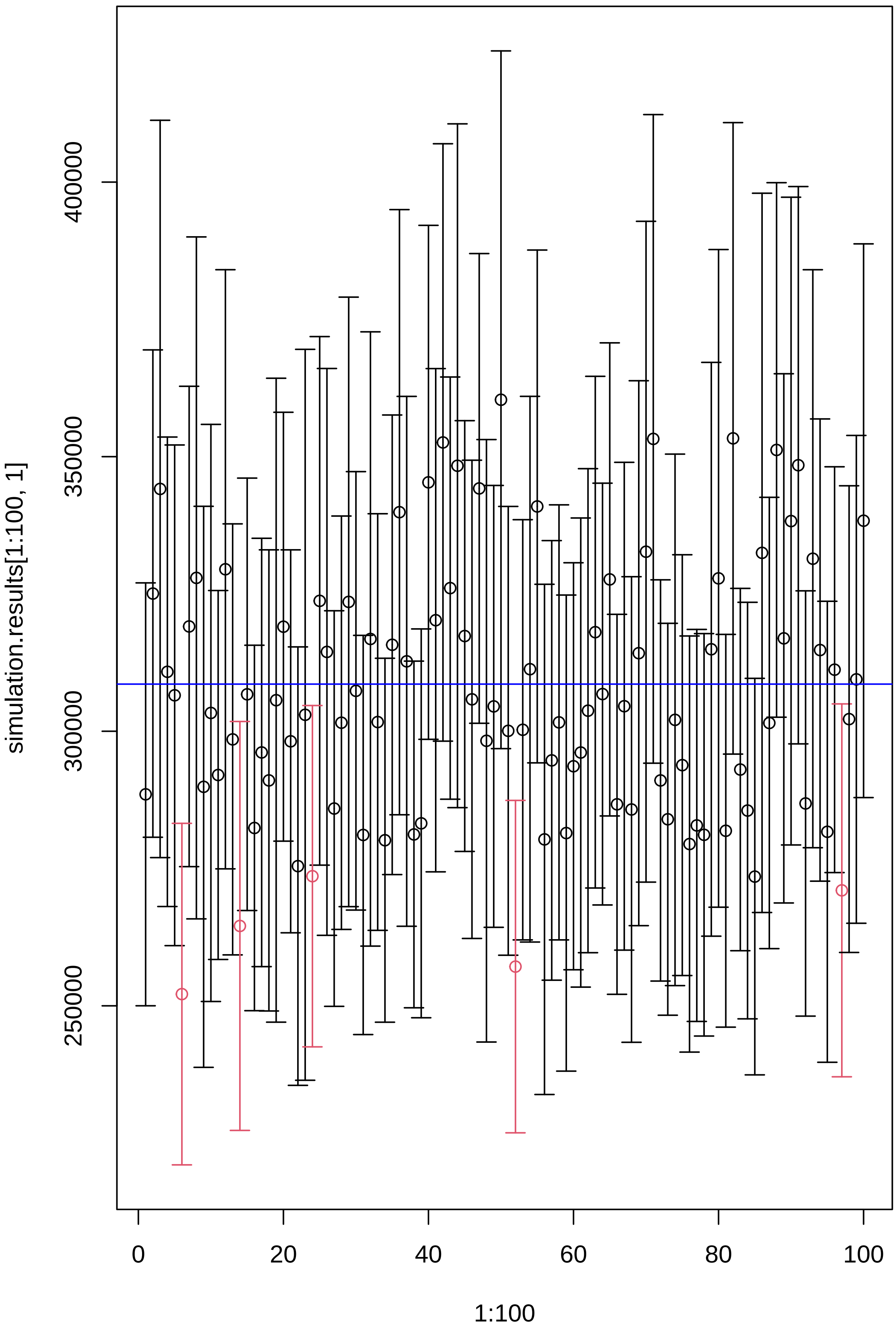
```
## [1] 23320.29
```

Empirical Coverage Rate of CIs

```r
simulation.results <- cbind (simulation.results, (simulation.results[,3] < ybarU) * (ybarU < simulation
colnames(simulation.results)[5] <- "Covered?"
head(simulation.results)
```

```
##          Est.     S.E.   ci.low   ci.upp Covered?
## [1,] 288511.1 19565.96 250006.7 327015.5        1
## [2,] 325070.6 22549.09 280695.6 369445.6        1
## [3,] 344123.0 34113.49 276990.0 411255.9        1
## [4,] 310826.7 21721.46 268080.4 353573.1        1
## [5,] 306543.3 23164.41 260957.3 352129.2        1
## [6,] 252130.9 15800.82 221036.0 283225.8        0
```

```r
library("plotrix")
par(mfrow=c(1,1))
plotCI(x=1:100,
       y=simulation.results[1:100,1],
       li = simulation.results[1:100,3],
       ui = simulation.results[1:100,4],
       col = 2-simulation.results[,5])
abline(h=ybarU, col = "blue")
```



simulation.results[1:100, 1]
250000
300000
350000
400000
0
20
40
60
80
100
1:100

```r
# Empirical coverage rate
mean (simulation.results[,"Covered?"])
```

```
## [1] 0.935
```