GR5234 - Final Exam

Mathieu Sauterey - UNI: mjs2364

December 10, 2018

Problem 1

Consider the population of size N = 50 given in the file FinalPop.csv, in the "Data" folder on Courseworks. Verify that the population mean and variance are y_bar.U = 14.882 and S2 = 0.977457, respectively.

```
# Loads data
population <- read.csv("FinalPop.csv", header = T)

# Population size
N = 50

mean(population$y)
```

```
## [1] 14.882
```

```
var(population$y)
```

```
## [1] 0.9774571
```

The output above shows that $\bar{y}_U$ **= 14.882 and S^2 = 0.977457**

(a) Find the design-based bias and MSE of the sample mean for an SRS of size n = 10, that is, find E(y_bar - y_bar.U) and E[(y_bar - y_bar.U)^2].

```
# Sample size
n = 10

# Variance of the design-based unbiased estimator
(1-n/N)*var(population$y)/n
```

```
## [1] 0.07819657
```

$\hat{\bar{y}}$ is an unbiased estimator of $\bar{y}_U$ so design-based bias = **0**

MSE = Bias^2 + Variance = 0^2 + Variance = **0.078197**

(b) Consider a stratified random sample of 2 units drawn from each of the 5 strata, defined as units 1-10, 11-20, 21-30, 31-40, and 41-50. Find the design-based bias and MSE of the stratified sample mean.

```
# Creates each strats
strat1 = population$y[1:10]
strat2 = population$y[11:20]
strat3 = population$y[21:30]
strat4 = population$y[31:40]
strat5 = population$y[41:50]
```

```
# Binds strats together
strat_data = list(strat1, strat2, strat3, strat4, strat5)

# Number of sample units in each strats
n.h = c(2, 2, 2, 2, 2)

# Number of population units in each strats
N.h = c(10, 10, 10, 10, 10)

# Population variance of strats
S.h = sapply(strat_data, sd)

# Variance of the stratified estimator
sum((N.h/N)^2 * S.h^2 / n.h * (1 - n.h/N.h))
```

```
## [1] 0.08310425
```

$\bar{y}_{strat}$ is an unbiased estimator of $\bar{y}_U$ so design-based bias = **0**

MSE = Bias^2 + Variance = 0^2 + Variance = **0.08310425**. Surprisingly the variance of the proportional stratified estimator is larger than the variance of the SRS estimator. This happens because the Nh are small (each Nh = 5).

Now suppose that the population Y1, Y2, . . . YN is itself a random sample of size N = 50 from a normally distributed superpopulation with mean $\mu = 15$ and variance $\sigma^2 = 1$.

(c) Find the model-based bias and MSE of the estimator in part (a)

```
sigma2 = 1

# Variance of the model-based unbiased SRS estimator is same as design-based
(1-n/N)*sigma2/n
```

```
## [1] 0.08
```

$\hat{\bar{y}}$ is an unbiased estimator of $\bar{y}_U$ so model-based bias = **0**

MSE = Bias^2 + Variance = 0^2 + Variance = **0.08**

(d) Find the model-based bias and MSE of the estimator in part (b)

```
# Variance of the model-based unbiased stratified estimator is same as design-based
sum((N.h/N)^2 * sigma2^2 / n.h * (1 - n.h/N.h))
```

```
## [1] 0.08
```

$\bar{y}_{strat}$ is an unbiased estimator of $\bar{y}_U$ so model-based bias = **0**

MSE = Bias^2 + Variance = 0^2 + Variance = **0.08**

Problem 2

The data for this problem are in the SDaA package. Data will contain total population and number of veterans for a random sample of 100 of the 3141 counties in the United States. The total population at the time of this data set was 255,077,536.

(a) Using ratio estimation, find an approximate 95% confidence interval for the total number of veterans in the United States. Report your answer in millions of veterans, rounded to the nearest 10,000.

```r
library(SDaA)
```

```
## Warning: package 'SDaA' was built under R version 3.4.4
```

```r
Data <- counties[,c(2,3,5,17)]

# Pop size
N = 3141
n = dim(Data)[1]
t_x = 255077536

# Sample data and pop of x
x = Data$totpop
y = Data$veterans
xbar.U = t_x / N

# Ratio estimator function
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)
}

# Calculates total number of veterans in the United States + SE
result = ratio.estimator.mean(x.samp = x, y.samp = y, N, xbar.U)

# 95% CI for the total number of veterans in the United States in millions
N * ( result[1] + c(-1,1) * 1.96 * result[2] ) / 1e6
```

```
## [1] 23.15261 29.56983
```

[**23.15 , 29.57**] is 95% confidence interval for the total number of veterans in the United States in millions.

(b) Using regression estimation, find an approximate 95% confidence interval for the total number of veterans in the United States. Report your answer in millions of veterans, rounded to the nearest 10,000.

```r
# Regression estimator function
regression.estimator.mean <- function(x.samp, y.samp, N, xbar.U)
{
  n <- length(y.samp)
  xbar <- mean(x.samp); ybar <- mean(y.samp);
  fit <- lsfit(x.samp, y.samp)
  B1.hat <- as.numeric(fit$coefficients)[2]
  ybar.hat.reg <- ybar + B1.hat * (xbar.U - xbar)
  e <- fit$residuals
  V.hat <- var(e)/n * (1 - n/N)
  SE <- sqrt(V.hat)
  answer <- c(point.est=ybar.hat.reg, std.error=SE)
  return(answer)
}

# Calculates the total number of veterans in the United States + SE
result <- regression.estimator.mean(x.samp = x, y.samp=y, N=N, xbar.U=xbar.U)

# 95% CI for the total number of veterans in the United States in millions
N * ( result[1] + c(-1,1) * 1.96 * result[2] ) / 1e6
```

```
## [1] 25.64022 30.11690
```

[**25.64** , **30.12**] is 95% confidence interval for the total number of veterans in the United States in millions.

(c) Assume the population values are themselves a random sample from a superpopulation in which Y_i = beta0 + beta1*x_i + epsilon_i where E(epsilon_i) = 0 and V(epsilon_i) = sigma^2. Find the model-based standard error of the estimate you computed for part (b). How does it compare to the design-based SE?

```r
# Model-based inference under "regression model"

# The model-based estimate is the same as design-based

# The model-based standard error:

Sxx <- (n-1) * var(x)

m1 <- lm(y ~ x)

sigma.hat <- sigma(m1)

SE_M.ybar.reg <- sigma.hat * sqrt(1/n + (xbar.U-mean(x))^2 / Sxx)

N * SE_M.ybar.reg
```

```
## [1] 1169519
```

$SE_M[\hat{t}_{yreg}]$ = **1,169,519** is the model-based standard error of the regression estimate. It is slightly higher than the design-based SE for part (b) which was equal to 1,142,010.

(d) Assume the population values are themselves a random sample from a superpopulation in which Y_i = beta1x_i + epsilon_i where E(epsilon_i) = 0 and V(epsilon_i) = x_i*sigma^2. Find the model-based standard error of the estimate you computed for part (a). How does it compare to the design-based SE?

```
# Model-based inference under regression through the origin model

m1 <- lm(y ~ 0 + x, weights=1/x)

# The model-based estimate of t_y = total veterans is same as design-based

# Model-based standard error is

sigma.hat <- sigma(m1)

SE_M.t_yr <- sigma.hat * t_x / sqrt(n*mean(x)) * sqrt(1 - (n*mean(x))/t_x)

SE_M.t_yr
```

```
## [1] 457516.8
```

$SE_M[\hat{t}_{yreg}]$ = **457,516.8** is the model-based standard error of the regression estimate. It is four times smaller than the design-based SE for part (b) which was equal to 1,637,046.

Problem 3

A fisherman is interested in N, the number of fish in a certain pond. He catches 100 fish, tags them, and throws them back. A few days later, he returns and catches 80 fish, of which 18 are tagged.

(a) Find the maximum likelihood estimate of N, along with its standard error.

```
# n1 units are captured and tagged (n1 fixed)
n1 <- 100

# n2 units are captured in follow-up (n2 fixed)
n2 <- 80

# Among the n2 captured m of them are tagged (m random)
m <- 18

# Confidence level & bootstrap  parameters
conf.level = 0.90
alpha <- 1 - conf.level

# Maximum likelihood estimate of N
```

```
N.hat <- n1 * n2 / m
N.hat
```

```
## [1] 444.4444
```

```
# Standard error of MLE of N
V.hat <- n1^2 * n2 * (n2 - m) / m^3
SE.N.hat <- sqrt(V.hat)
SE.N.hat
```

```
## [1] 92.22148
```

The maximum likelihood estimate of N is **444.44** and its standard error is **92.22**

(b) Explain in plain English what your answer to part (a) means, particularly the standard error. The fisherman does not know or care about maximum likelihood theory, or sampling distributions, or any such things - he just wants to know how many fish are in the pond. Help him.

The estimated number of fish in the pond is 444 (rounded to the nearest integer). The number of fish by which this estimate is likely to be off is roughly 92.

(c) Find an approximate 90% confidence interval for N by inverting the acceptance region of a level .10 Pearson's chi-square test for independence between the binary variables In first day's catch and In second day's catch.

```
# Better way to do CI: Acceptance region for chi-square test
x11 <- m
x12 <- n1 - m
x21 <- n2 - m
x22.hat <- round(N.hat - x11 - x12 - x21);

# Creates table
X <- matrix(c(x11,x21,x12,x22.hat),2,2)

# Chi-square test
chisq.test(X)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  X
## X-squared = 6.4809e-30, df = 1, p-value = 1
```

```
chisq.test(X)$p.value
```

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

```
# First find the lower bound
Reject <- FALSE
u <- x22.hat;

while(!Reject)
{
 X <- matrix(c(x11,x21,x12,u),2,2)
 Reject <- chisq.test(X, correct=F)$p.value < alpha
 u <- u - 1
}

u <- u + 2
N.lower <- x11 + x12 + x21 + u


# Now the upper bound:
u <- x22.hat
Reject <- FALSE;

while(!Reject)
{
 X <- matrix(c(x11,x21,x12,u),2,2)
 Reject <- chisq.test(X, correct=F)$p.value < alpha
 u <- u + 1
}

u <- u - 2
N.upper <- x11 + x12 + x21 + u

# CI
c(N.lower=N.lower, N.upper=N.upper)
```

```
## N.lower N.upper
##     336     618
```

[**336 , 618**] is an approximate 90% confidence interval for N using an invertion of the acceptance region for chi-square test.

Problem 4

The file statepop.csv, available in the “Data” folder on Courseworks, lists the 1992 population for the 50 states plus the District of Columbia. The file counties.csv contains the number of counties for a sample of size 12 with replacement, with probabilities proportional to population.

(a) Estimate the total number of counties in the United States, and find the standard error of your estimate.

```
# Loads data
state <- read.csv("statepop.csv", header = T)
county <- read.csv("counties.csv", header = T)
sample <- merge(state, county)

# Total 1992 US population
M.0 <- sum(state$popn)

# States' population
```

```
M.i <- sample$popn

# probabilities proportional to population
psi.i <- M.i / M.0

# Sample number of counties
t.i <- sample$counties

# Estimated number of US counties
t.hat.psi <- mean(t.i/psi.i)
t.hat.psi
```

```
## [1] 2353.318
```

```
# Standard error?

n <- dim(county)[1]
V.hat <- 1/n * var(t.i/psi.i)
SE <- sqrt(V.hat)
SE
```

```
## [1] 648.8684
```

$\mathbf{E}[\hat{t}_{\psi}] = \mathbf{2,353.32}$

$\mathbf{SE}[\hat{t}_{\psi}] = \mathbf{648.87}$

(b) With California being sampled three times and New Jersey twice, there were nine distinct states in the sample. Writing your estimate in part (a) as t_hat = Sum[(i in R) w_i x Q_i x t_i] with R = {CA, CO, CT, MA, MO, NJ, TN, VA, WI} and t_i = number of counties in state i, find w_i x Q_i for each of those nine states. What is Sum[(i in R) w_i x Q_i] for this sample? What is its expected value over repeated random sampling?

```
n <- 9

# number of times unit i is counted in the sample
Q.i <- table(county$state)

# States' population
M.i <- unique(sample)$popn

# probabilities proportional to population
psi.i <- M.i / M.0

# sample weights
w.i = 1/(n*psi.i)

sum(Q.i*w.i)
```

```
## [1] 52.76213
```

```
Q.i*w.i
```

```
##
##    California      Colorado   Connecticut Massachusetts      Missouri
##      2.752055      8.180248      8.643153      4.729395      5.460111
##    New Jersey     Tennessee      Virginia     Wisconsin
##      7.248327      5.639887      4.432244      5.676709
```

w_iQ_i **for each of those 9 states is shown in the output above.**

$\sum_{i \in R^*}(w_iQ_i) =$ **52.76**. Its expected value over repeated random sampling is **52.0**

Problem 5

Investigators selected a random sample of 200 teenagers from a population of 2000 for a survey of screen time on smartphones and other handheld devices; the overall response rate was 75%. A follow-up sample was taken of 10 of the 50 nonrespondents, with responses obtained from all 10. In the data file ScreenTime.csv, the variable Group takes the value 1 for respondents, 2 for nonrespondents included in the follow-up survey, and 3 for nonrespondents not in the follow-up sample; the variable Minutes gives that individual's average daily screen time in minutes.

Give an approximate 95% confidence interval for the average screen time per day among these 2000 teenagers.

```
# Loads data
data <- read.csv("ScreenTime.csv", header = T)

# Population size
N <- 2000

# Sample size
n <- dim(data)[1]

# Number of respondents
n.R <- sum(data$Group == 1)

# Number of non-respondents
n.M <- sum(data$Group != 1)

# Sample mean of respondents
y_bar.R <- mean(subset(data, Group == 1)[,"Minutes"])
V.R <- var(subset(data, Group == 1)[,"Minutes"])

# Sample mean of call-backed respondents
y_bar.M <-  mean(subset(data, Group == 2)[,"Minutes"])
V.M <- var(subset(data, Group == 2)[,"Minutes"])

# Estimated Population mean
y_bar.hat <- n.R*y_bar.R/n + n.M*y_bar.M/n

# Variance of estimator (0.2 is call back ratio = 10/50)
```

```r
V.y_bar.hat <- (n.R-1)*V.R/(n*(n-1)) + (n.M-1)*V.M/(n.M*0.2*(n-1)) +
                (1/(n-1))*((n.R*(y_bar.R-y_bar.hat)^2/n) + (n.M*(y_bar.M-y_bar.hat)^2/n))

# Standard error of estimator
SE.y_bar.hat <- sqrt(V.y_bar.hat)

# 95% CI
y_bar.hat + c(-1,1)*1.96*SE.y_bar.hat
```

```
## [1] 147.6742 242.8658
```

[**147.67, 242.87**] is an approximate 95% confidence interval for the average screen time (in minutes) per day among these 2000 teenagers.