MATH3426 - Midterm Project

Wilbert Caine

```r
#change seed 20584260
#find a way to make sure that ntot=sum of rounding
set.seed(20387844)
data <- read.csv("Big_Five_Personality_small.csv")
head(data)
```

```
##   X EXT1 EXT2 EST1 EST2 AGR1 AGR2 CSN1 CSN2            dateload country
## 1 1    1    5    4    3    2    5    4    4 2016-03-03 02:11:06      AU
## 2 2    1    5    5    1    1    3    4    2 2016-03-03 02:22:56      AU
## 3 3    1    3    3    2    5    4    4    4 2016-03-03 02:26:37      AU
## 4 4    3    4    1    2    1    4    4    2 2016-03-03 02:43:01      AU
## 5 5    4    1    3    4    1    4    4    4 2016-03-03 03:11:20      AU
## 6 6    4    2    1    5    1    5    5    2 2016-03-03 03:26:32      AU
##   lat_appx_lots_of_err long_appx_lots_of_err
## 1             -37.9333              145.2333
## 2             -27.0000              133.0000
## 3             -31.6448              152.7946
## 4             -27.0000              133.0000
## 5             -27.0000              133.0000
## 6             -37.8139              144.9634
```

Part I. Simple Random Sampling

We first estimate the population mean of variable "EXT1".

1. Find the population mean and population total of "EXT1".

```r
mu_ext1 <- mean(data$EXT1)
N <- nrow(data)
tau_ext1 <- mu_ext1 * N
print(list(
  "population mean of "EXT1"" = mu_ext1,
  "population total of "EXT1"" = tau_ext1
))
```

```
## $`population mean of "EXT1"`
## [1] 2.64419
##
## $`population total of "EXT1"`
## [1] 886938
```

2. Use simple random sampling with n=1500 to estimate the population mean and total of "EXT1" respectively. Construct 96% confidence interval for the population mean and total of "EXT1". What are the error bounds for these estimates?

```r
n <- 15e2
N <- nrow(data)
alpha <- 0.04
sample_ext1 <- sample(data$EXT1, n)

ybar_ext1 <- mean(sample_ext1)
s_square_ext1 <- var(sample_ext1)
varhat_ybar_ext1 <- (s_square_ext1 / n) * (1 - n / N)
errorbound_muhat_ext1 = qnorm(1 - alpha / 2) * sqrt(varhat_ybar_ext1)

tauhat_ext1 <- N * ybar_ext1
varhat_tauhat_ext1 <- (N ^ 2) * varhat_ybar_ext1
errorbound_tauhat_ext1 = qnorm(1 - alpha / 2) * sqrt(varhat_tauhat_ext1)

print(
  list(
    "estimate of population mean" = ybar_ext1,
    "estimate of population mean" = tauhat_ext1,
    "error bound for estimate of population mean" = errorbound_muhat_ext1,
    "error bound for estimate of population mean" = errorbound_tauhat_ext1,
    "CI.lower for mu" = ybar_ext1 - errorbound_muhat_ext1,
    "CI.lower for tau" = tauhat_ext1 - errorbound_tauhat_ext1,
    "CI.upper for mu" = ybar_ext1 + errorbound_muhat_ext1,
    "CI.upper for tau" = tauhat_ext1 + errorbound_tauhat_ext1
  )
)
```

```
## $`estimate of population mean`
## [1] 2.702
##
## $`estimate of population mean`
## [1] 906329.2
##
## $`error bound for estimate of population mean`
## [1] 0.06672092
##
## $`error bound for estimate of population mean`
## [1] 22380.13
##
## $`CI.lower for mu`
## [1] 2.635279
##
## $`CI.lower for tau`
## [1] 883949
##
## $`CI.upper for mu`
## [1] 2.768721
##
## $`CI.upper for tau`
## [1] 928709.3
```

3. Repeat step 2 for 100 times and calculate the relative frequencies that the constructed confidence intervals contain the population mean and total, respectively. Are the relative frequencies close to 96%?

```r
CI_cal_mu_ext1 <- function(sample, n, N) {
  ybar_ext1 <- mean(sample)
  varhat_ybar_ext1 <- (var(sample) / n) * (1 - n / N)
  errorbound_muhat_ext1 = qnorm(1 - alpha / 2) * sqrt(varhat_ybar_ext1)
  list(
    "CI.lower for mu" = ybar_ext1 - errorbound_muhat_ext1,
    "CI.upper for mu" = ybar_ext1 + errorbound_muhat_ext1,
    "error bound for mu" = errorbound_muhat_ext1
  )
}
CI_cal_tau_ext1 <- function(sample, n, N) {
  ybar_ext1 <- mean(sample)
  varhat_ybar_ext1 <- (var(sample) / n) * (1 - n / N)

  tauhat_ext1 <- N * ybar_ext1
  varhat_tauhat_ext1 <- (N ^ 2) * varhat_ybar_ext1
  errorbound_tauhat_ext1 = qnorm(1 - alpha / 2) * sqrt(varhat_tauhat_ext1)
  list(
    "CI.lower for tau" = tauhat_ext1 - errorbound_tauhat_ext1,
    "CI.upper for tau" = tauhat_ext1 + errorbound_tauhat_ext1,
    "error bound for tau" = errorbound_tauhat_ext1
  )
}
mrep <- 1e2
temp_mu_ext1 <- logical(mrep)
temp_tau_ext1 <- logical(mrep)
for (i in 1:mrep) {
  sample_ext1 <- sample(data$EXT1, n)
  CI_mu <- CI_cal_mu_ext1(sample_ext1, n, N)
  temp_mu_ext1[i] <- (CI_mu[[1]] <= mu_ext1) * (CI_mu[[2]] >= mu_ext1)
  CI_tau <- CI_cal_tau_ext1(sample_ext1, n, N)
  temp_tau_ext1[i] <-
    (CI_tau[[1]] <= tau_ext1) * (CI_tau[[2]] >= tau_ext1)
}
print(
  list(
    "relative frequencies that the constructed confidence intervals contain the population mean" = mean
    "relative frequencies that the constructed confidence intervals contain the population total" = mean
  )
)
```

```
## $`relative frequencies that the constructed confidence intervals contain the population mean`
## [1] 0.92
##
## $`relative frequencies that the constructed confidence intervals contain the population total`
## [1] 0.92
```

4. Use the sample variance from step 2, calculate the the sample size lower bound for estimating the population mean of "EXT1" with an error bound 0.015, at the significance level 0.05.

```r
alpha <- 0.05
V_ext1 <- (0.015 / qnorm(1 - alpha / 2)) ^ 2
sample_size_lower_ext1 <- (N * s_square_ext1) / (N * V_ext1 + s_square_ext1)
sample_size_lower_ext1
```

```
## [1] 25117.59
```

5. Use the sample size lower bound from step 4, random sample this number of observations and calculate the error bound at the significance level 0.05, is it close to 0.015? Repeat this process 100 times and calculate the relative frequency that the error bound is smaller than 0.015.

```r
n <- ceiling(sample_size_lower_ext1)
alpha <- 0.05
sample_ext1 <- sample(data$EXT1, n)
s_square_ext1 <- var(sample_ext1)
varhat_ybar_ext1 <- (s_square_ext1 / n) * (1 - n / N)
errorbound_muhat_ext1 = qnorm(1 - alpha / 2) * sqrt(varhat_ybar_ext1)
print(
  list(
    "error bound for random sample with the sample size lower bound from step 4" = errorbound_muhat_ext
  )
)
```

```
## $`error bound for random sample with the sample size lower bound from step 4`
## [1] 0.01492682
```

```r
mrep <- 1e2
temp_mu_ext1 <- logical(mrep)
for (i in 1:mrep) {
  sample_ext1 <- sample(data$EXT1, n)
  s_square_ext1 <- var(sample_ext1)
  varhat_ybar_ext1 <- (s_square_ext1 / n) * (1 - n / N)
  errorbound_muhat_ext1 = qnorm(1 - alpha / 2) * sqrt(varhat_ybar_ext1)
  temp_mu_ext1[i] <- errorbound_muhat_ext1 < 0.015
}
print(list(
  "relative frequency that the error bound is smaller than 0.015" = mean(temp_mu_ext1)
))
```

```
## $`relative frequency that the error bound is smaller than 0.015`
## [1] 0.88
```

We then estimate the population proportion of answer "5" for variable "EXT2".

6. For the variable "EXT2", what is the population proportion of answer "5"?

```r
N <- nrow(data)
p_ext2.5 <- mean(data$EXT2 == 5)
print(list("population proportion of answer "5" for variable "EXT2"" = p_ext2.5))
```

```
## $`population proportion of answer "5" for variable "EXT2"`
## [1] 0.1170113
```

7. Use simple random sampling with n=5000 to estimate the population proportion of answer "5" for variable "EXT2". What is the confidence interval and error bound of your estimate at significance level 0.05 ?

```r
n <- 5e3
N <- nrow(data)
alpha <- 0.05
sample_ext2 <- sample(data$EXT2, n)
```

```r
phat_ext2.5 <- mean(sample_ext2 == 5)
varhat_phat_ext2.5 <- (1 - n / N) * (phat_ext2.5 * (1 - phat_ext2.5)) /
  (n - 1)
errorbound_phat_ext2.5 = qnorm(1 - alpha / 2) * sqrt(varhat_phat_ext2.5)

print(
  list(
    "estimate of the population proportion of answer "5" for variable "EXT2"" = phat_ext2.5,
    "error bound for estimate of the population proportion" = errorbound_phat_ext2.5,
    "CI.lower for p" = phat_ext2.5 - errorbound_phat_ext2.5,
    "CI.upper for p" = phat_ext2.5 + errorbound_phat_ext2.5
  )
)
```

```
## $`estimate of the population proportion of answer "5" for variable "EXT2"`
## [1] 0.1132
##
## $`error bound for estimate of the population proportion`
## [1] 0.008717287
##
## $`CI.lower for p`
## [1] 0.1044827
##
## $`CI.upper for p`
## [1] 0.1219173
```

8. Repeat step 7 for 100 times and calculate the relative frequencies that the confidence intervals contain the population proportion of answer “5” for the variable “EXT2”.

```r
CI_cal_p_ext2.5 <- function(sample, n, N) {
  phat_ext2.5 <- mean(sample == 5)
  varhat_phat_ext2.5 <- (1 - n / N) * (phat_ext2.5 * (1 - phat_ext2.5)) /
    (n - 1)
  errorbound_phat_ext2.5 = qnorm(1 - alpha / 2) * sqrt(varhat_phat_ext2.5)
  list(
    "CI.lower for p" = phat_ext2.5 - errorbound_phat_ext2.5,
    "CI.upper for p" = phat_ext2.5 + errorbound_phat_ext2.5,
    "error bound for p" = errorbound_phat_ext2.5
  )
}
mrep <- 1e2
temp_p_ext2.5 <- logical(mrep)
for (i in 1:mrep) {
  sample_ext2.5 <- sample(data$EXT2, n)
  CI_p <- CI_cal_p_ext2.5(sample_ext2.5, n, N)
  temp_p_ext2.5[i] <- (CI_p[[1]] <= p_ext2.5) * (CI_p[[2]] >= p_ext2.5)
}
print(
  list(
    "relative frequencies that the constructed confidence intervals contain the population proportion"
  )
)
```

```
## $`relative frequencies that the constructed confidence intervals contain the population proportion`
## [1] 0.93
```

9. Use the sample variance from step 7, calculate the sample size lower bound for estimating the population proportion of answer "5" of variable "EXT2" with an error bound of 0.02, at the significance level 0.04.

```r
alpha <- 0.04
V_ext2 <- (0.02 / qnorm(1 - alpha / 2)) ^ 2
sample_size_lower_ext2.5 <-
  (N * phat_ext2.5 * (1 - phat_ext2.5)) / (N * V_ext2 + phat_ext2.5 * (1 -
                                                                        phat_ext2.5))
sample_size_lower_ext2.5
```

```
## [1] 1055.209
```

Part II. Stratified Simple Random Sampling

By looking at the country code of all participants in the provided dataset, the participants come from 6 different countries "AU", "CA", "DE", "GB", "US", "PH" In this part, we implement stratified SRS to estimate population parameters for the variables "EST1" and "EST2".

We first estimate the population mean of variable "EST1".

1. Find the population sizes of the 6 strata and population mean of "EST1".

```r
N_country <- summary(data$country)
mu_est1 <- mean(data$EST1)
print(list(
  "population size of the 6 strata" = N_country,
  "population mean of "EST1"" = mu_est1
))
```

```
## $`population size of the 6 strata`
##     AU     CA     DE     GB     PH     US
##  49753  61805  14084  66487  19844 123456
##
## $`population mean of "EST1"`
## [1] 3.289444
```

2. To estimate the population mean of "EST1", use simple random sampling to sample 400 from 'AU', 600 from 'CA', 140 from 'DE', 660 from 'GB', 200 from 'PH' and 1200 from 'US'. Compute your point estimate and error bound at significance level 0.05. Construct the confidence interval.

```r
n_country <- c(400, 600, 140, 660, 200, 1200)
country <- c('AU', 'CA', 'DE', 'GB', 'PH', 'US')
alpha <- 0.05
sample_est1 <- list()
for (i in 1:6) {
  sample_est1[[i]] <-
    sample(data$EST1[data$country == country[i]], n_country[i])
}

ybar_est1 <- sapply(sample_est1, mean)
s_square_est1 <- sapply(sample_est1, var)
W_country <- N_country / sum(N_country)
yst_est1 <- sum(W_country * ybar_est1)
varhat_ybar_est1 <-
  sum(W_country ^ 2 * (1 - n_country / N_country) * (s_square_est1 / n_country))
```

```r
err_bound_muhat_est1 = qnorm(1 - alpha / 2) * sqrt(varhat_ybar_est1)
print(
  list(
    "estimate of population mean of "EST1"" = yst_est1,
    "error bound for estimate of population mean" = err_bound_muhat_est1,
    "CI.lower for mu" = yst_est1 - err_bound_muhat_est1,
    "CI.upper for mu" = yst_est1 + err_bound_muhat_est1
  )
)
```

```
## $`estimate of population mean of "EST1"`
## [1] 3.328766
##
## $`error bound for estimate of population mean`
## [1] 0.04569977
##
## $`CI.lower for mu`
## [1] 3.283066
##
## $`CI.upper for mu`
## [1] 3.374466
```

3. Repeat step 2 for 100 times and calculate the relative frequencies that the constructed confidence intervals contain the population mean of "EST1". Is the relative frequency close to 95%?

```r
CI_cal_mu_est1 <- function(sample, n, N) {
  ybar <- sapply(sample, mean)
  s_square <- sapply(sample, var)
  W <- N / sum(N)
  yst <- sum(W * ybar)
  varhat_ybar <- sum(W ^ 2 * (1 - n / N) * (s_square / n))
  err_bound_muhat = qnorm(1 - alpha / 2) * sqrt(varhat_ybar)
  list(
    "CI.lower for mu" = yst - err_bound_muhat,
    "CI.upper for mu" = yst + err_bound_muhat,
    "error bound for mu" = err_bound_muhat
  )
}

mrep <- 1e2
temp_mu_est1 <- logical(mrep)
for (i in 1:mrep) {
  sample_est1 <- list()
  for (j in 1:6) {
    sample_est1[[j]] <-
      sample(data$EST1[data$country == country[j]], n_country[j])
  }
  CI_mu <- CI_cal_mu_est1(sample_est1, n_country, N_country)
  temp_mu_est1[i] <- (CI_mu[[1]] <= mu_est1) * (CI_mu[[2]] >= mu_est1)
}
print(
  list(
    "relative frequencies that the constructed confidence intervals contain the population mean" = mean
  )
)
```

```
## $`relative frequencies that the constructed confidence intervals contain the population mean`
## [1] 0.95
```

4. Use the sample variances and allocation weights from step 2, calculate the the sample size lower bound for estimating the population mean of "EST1" with an error bound 0.02, at the significance level 0.05.

```
alpha <- 0.05
w_country <- n_country / sum(n_country)
sample_size_lower_est1 <-
  (sum((N_country ^ 2) * s_square_est1 / w_country)) / ((sum(N_country) *
                                                          0.02 / qnorm(1 - alpha / 2)) ^ 2 + sum(N_cou
sample_size_lower_est1
```

```
## [1] 16064.93
```

5. Use the sample size lower bound from step 4 and allocating weights as step 2, randomly sample these numbers of observations and calculate the error bound at the significance level 0.05, is it close to 0.02? Repeat this process 100 times and calculate the relative frequency that the error bound is smaller than 0.02.

```
n_country_lower_est1 <-
  round(ceiling(sample_size_lower_est1) * w_country)
sample_est1 <- list()
for (i in 1:6) {
  sample_est1[[i]] <-
    sample(data$EST1[data$country == country[i]], n_country_lower_est1[i])
}
CI_mu <-
  CI_cal_mu_est1(sample_est1, n_country_lower_est1, N_country)
print(list("error bound for random sample with the sample size lower bound" = CI_mu[[3]]))
```

```
## $`error bound for random sample with the sample size lower bound`
## [1] 0.02021361
```

```
mrep <- 1e2
temp <- logical(mrep)
for (i in 1:mrep) {
  sample_est1 <- list()
  for (j in 1:6) {
    sample_est1[[j]] <-
      sample(data$EST1[data$country == country[j]], n_country[j])
  }
  CI_mu <- CI_cal_mu_est1(sample_est1, n_country, N_country)
  temp_mu_est1[i] <- CI_mu[[3]] < 0.02
}
print(list(
  "relative frequency that the error bound is smaller than 0.02" = mean(temp_mu_est1)
))
```

```
## $`relative frequency that the error bound is smaller than 0.02`
## [1] 0
```

We then estimate the population proportion of answer "5" for variable "EST2".

6. Find population proportion of answer "5" for variable "EST2".

```r
N <- nrow(data)
p_est2.5 <- mean(data$EST2 == 5)
print(list("population proportion of answer "5" for variable "EST2"" = p_est2.5))
```

```
## $`population proportion of answer "5" for variable "EST2"`
## [1] 0.149668
```

7. Use stratified simple random sampling with n=5000 and Neyman allocation to estimate the population proportion of answer "5" for the variable "EST2". What is the confidence interval and error bound of your estimate at significance level 0.05 ?

```r
alpha <- 0.05
phat_country <- numeric()
s_square_country <- numeric()
for (i in 1:6) {
  phat_country[[i]] <- mean(data$EST2[data$country == country[i]] == 5)
  s_square_country[[i]] <- phat_country[[i]] * (1 - phat_country[[i]])
}
w_country <-
  N_country * sqrt(s_square_country) / sum(N_country * sqrt(s_square_country))
n_country <- round(5000 * w_country)

phat_est2.5 <- numeric()
s_square_phat_est2.5 <- numeric()
for (i in 1:6) {
  phat_est2.5[[i]] <-
    mean(sample(data$EST2[data$country == country[i]], n_country[i]) == 5)
  s_square_phat_est2.5[[i]] <- phat_est2.5[[i]] * (1 - phat_est2.5[[i]])
}
pst_est2.5 <- sum(N_country * phat_est2.5) / sum(N_country)
var_pst_est2.5 <-
  sum((N_country ^ 2) * (1 - n_country / N_country) * s_square_phat_est2.5 /
        (n_country - 1)) / (sum(N_country) ^ 2)
err_bound_pst_est2.5 <- qnorm(1 - alpha / 2) * sqrt(var_pst_est2.5)
print(
  list(
    "estimate of population proportion of answer "5"" = pst_est2.5,
    "error bound for estimate of population proportion" = err_bound_pst_est2.5,
    "CI.lower for p" = pst_est2.5 - err_bound_pst_est2.5,
    "CI.upper for p" = pst_est2.5 + err_bound_pst_est2.5
  )
)
```

```
## $`estimate of population proportion of answer "5"`
## [1] 0.1494792
##
## $`error bound for estimate of population proportion`
## [1] 0.009811492
##
## $`CI.lower for p`
## [1] 0.1396677
##
## $`CI.upper for p`
## [1] 0.1592907
```

8. Repeat step 7 for 100 times and calculate the relative frequencies that the confidence intervals contain

the population proportion of answer “5” for the variable “EST2”.

```
mrep <- 1e2
temp_phat_est2.5 <- logical(mrep)
for (i in 1:mrep) {
  phat_est2.5 <- numeric()
  s_square_phat_est2.5 <- numeric()
  for (j in 1:6) {
    phat_est2.5[[j]] <-
      mean(sample(data$EST2[data$country == country[j]], n_country[[j]]) == 5)
    s_square_phat_est2.5[[j]] <-
      phat_est2.5[[j]] * (1 - phat_est2.5[[j]])
  }
  pst_est2.5 <- sum(N_country * phat_est2.5) / sum(N_country)
  var_pst_est2.5 <-
    sum((N_country ^ 2) * (1 - n_country / N_country) * s_square_phat_est2.5 /
          (n_country - 1)) / ((sum(N_country)) ^ 2)
  err_bound_pst_est2.5 <-
    qnorm(1 - alpha / 2) * sqrt(var_pst_est2.5)
  CI.lower <- pst_est2.5 - err_bound_pst_est2.5
  CI.upper <- pst_est2.5 + err_bound_pst_est2.5
  temp_phat_est2.5[i] <- (CI.lower <= p_est2.5) * (CI.upper >= p_est2.5)
}
print(
  list(
    "relative frequencies that the constructed confidence intervals contain the population proportion" =
  )
)
```

```
## $`relative frequencies that the constructed confidence intervals contain the population proportion`
## [1] 0.92
```

9. Use the sample variances from step 7 with Neyman allocation, calculate the sample size lower bound for estimating the population proportion of answer “5” of variable “EST2” with an error bound of 0.02, at the significance level 0.05.

```
sample_size_lower_est2 <-
  sum(N_country * sqrt(s_square_phat_est2.5)) ^ 2 / ((sum(N_country) * 0.02 /
                                                        qnorm(1 - alpha / 2)) ^ 2 + sum(N_country * s_s
sample_size_lower_est2
```

```
## [1] 1239.846
```

10. Repeat step 9, but use proportional allocation.

```
w_country <- N_country / sum(N_country)
n_country <- round(5000 * w_country)
phat_est2.5 <- numeric()
s_square_phat_est2.5 <- numeric()
for (i in 1:6) {
  phat_est2.5[[i]] <-
    mean(sample(data$EST2[data$country == country[i]], n_country[i]) == 5)
  s_square_phat_est2.5[[i]] <- phat_est2.5[[i]] * (1 - phat_est2.5[[i]])
}
sample_size_lower_est2 <-
  sum(N_country * s_square_phat_est2.5) / (
    sum(N_country) * (0.02 / qnorm(1 - alpha / 2)) ^ 2 + sum(N_country * s_square_phat_est2.5) /
```

```
      sum(N_country)
  )
sample_size_lower_est2
```

```
## [1] 1190.844
```

11. Use the sample size obtained from step 10 and with proportional allocation, randomly sample this number of observations from the population by stratified SRS and calculate the error bound. Repeat the process 100 times and calculate the relative frequencies that the error bound is less than 0.02.

```
n_country_lower_est2 <-
  round(ceiling(sample_size_lower_est2) * N_country / sum(N_country))
phat_est2.5 <- numeric()
s_square_phat_est2.5 <- numeric()
for (i in 1:6) {
  phat_est2.5[[i]] <-
    mean(sample(data$EST2[data$country == country[i]], n_country_lower_est2[i]) ==
           5)
  s_square_phat_est2.5[[i]] <- phat_est2.5[[i]] * (1 - phat_est2.5[[i]])
}
pst_est2.5 <- sum(N_country * phat_est2.5) / sum(N_country)
var_pst_est2.5 <-
  sum((N_country ^ 2) * (1 - n_country_lower_est2 / N_country) * s_square_phat_est2.5 /
        (n_country_lower_est2 - 1)
  ) / (sum(N_country) ^ 2)
err_bound_pst_est2.5 <- qnorm(1 - alpha / 2) * sqrt(var_pst_est2.5)
print(
  list("error bound for random sample with the sample size lower bound" = err_bound_pst_est2.5)
)
```

```
## $`error bound for random sample with the sample size lower bound`
## [1] 0.01992166
```

```
mrep <- 1e2
temp_p_est2 <- logical(mrep)
for (i in 1:mrep) {
  phat_est2.5 <- numeric()
  s_square_phat_est2.5 <- numeric()
  for (j in 1:6) {
    phat_est2.5[[j]] <-
      mean(sample(data$EST2[data$country == country[j]], n_country_lower_est2[j]) ==  5)
    s_square_phat_est2.5[[j]] <-
      phat_est2.5[[j]] * (1 - phat_est2.5[[j]])
  }
  pst_est2.5 <- sum(N_country * phat_est2.5) / sum(N_country)
  var_pst_est2.5 <-
    sum((N_country ^ 2) * (1 - n_country_lower_est2 / N_country) * s_square_phat_est2.5 / (n_country_lo
    ) / (sum(N_country) ^ 2)
  err_bound_pst_est2.5 <-
    qnorm(1 - alpha / 2) * sqrt(var_pst_est2.5)
  temp_p_est2[i] <- err_bound_pst_est2.5 < 0.02
}
print(list(
  "relative frequency that the error bound is smaller than 0.02" = mean(temp_p_est2)
))
```

```
## $`relative frequency that the error bound is smaller than 0.02`
## [1] 0.32
```

Part III. Systematic Random Sampling

Now, we apply systematic random sampling to estimate the population mean of variable "AGR1".

We estimate the population mean of variable "AGR1".

1. Find the population mean of variable "AGR1".

```r
mu_agr1 <- mean(data$AGR1)
print(list("population mean of "AGR1"" = mu_agr1))
```

```
## $`population mean of "AGR1"`
## [1] 2.207406
```

2. Choose k=100 and do a systematic random sampling. Calculate the point estimate and its error bound at significance level 0.05. Use half-sample estimator to estimate the required variance.

```r
sy <- function(data, sample_size, k, start_point) {
  sample = numeric()
  for (i in 1:sample_size) {
    sample[i] <- data[start_point + (i - 1) * k]
  }
  sample
}

N <- nrow(data)
k <- 100
alpha <- 0.05
start_point <- sample(1:k, 1)
n <- 2 * as.integer(N / (2 * k))
ybar_i1 <- mean(sy(data$AGR1, n / 2, 2 * k, start_point))
ybar_i2 <- mean(sy(data$AGR1, n / 2, 2 * k, start_point + k))
ybar_syhs <- mean(c(ybar_i1, ybar_i2))
s.2_ybar <- (ybar_i1 - ybar_i2) ^ 2 / 2
varhat_muhat_syhs <- (1 - n / N) * s.2_ybar / 2
error_bound <- qnorm(1 - alpha / 2) * sqrt(varhat_muhat_syhs)

print(
  list(
    "point estimate" = ybar_syhs,
    "error bound" = error_bound,
    "variance" = varhat_muhat_syhs
  )
)
```

```
## $`point estimate`
## [1] 2.228682
##
## $`error bound`
## [1] 0.01104731
##
```

```
## $variance
## [1] 3.176996e-05
```

3. Choose k=100 and do a systematic random sampling. Calculate the point estimate and its error bound at significance level 0.05. Use successive differences estimator to estimate the required variance.

```
N <- nrow(data)
k <- 100
alpha <- 0.05
start_point <- sample(1:k, 1)
n <- 2 * as.integer(N / (2 * k))
sample <- sy(data$AGR1, n, k, start_point)
sd.2 <- 0
for (i in 1:(n - 1)) {
  sd.2 <- sd.2 + (sample[i] - sample[i + 1]) ^ 2 / (n - 1)
}
ybar_sysd <- mean(sample)
varhat_muhat_sysd <- (1 - n / N) * (1 / 2) * sd.2 / n
error_bound <- qnorm(1 - alpha / 2) * sqrt(varhat_muhat_sysd)

print(
  list(
    "point estimate" = ybar_sysd,
    "error bound" = error_bound,
    "variance" = varhat_muhat_sysd
  )
)
```

```
## $`point estimate`
## [1] 2.213775
##
## $`error bound`
## [1] 0.04340054
##
## $variance
## [1] 0.0004903364
```

4. Choose k=100 and do a systematic random sampling. Calculate the point estimate and its error bound at significance level 0.05. Use 10 repeated system samples to estimate the required variance.

```
N <- nrow(data)
k <- 100
alpha <- 0.05
ns = 10
k_prime = k * ns
start_vec = sample(1:k_prime, ns)
sample_mat = matrix(NA, nrow = ns, ncol = n / ns)
for (i in 1:ns) {
  sample_mat[i, ] = sy(data$AGR1, n / ns, k_prime, start_vec[i])
}
ybar <- rowMeans(sample_mat)
ybar_syrs <- mean(ybar)
s.2_ybar <- var(ybar)
varhat_ybar_syrs <- (1 - n / N) * s.2_ybar / ns
error_bound = qnorm(1 - alpha / 2) * sqrt(varhat_ybar_syrs)
```

```r
print(
  list(
    "point estimate" = ybar_syrs,
    "error bound" = error_bound,
    "variance" = varhat_ybar_syrs
  )
)
```

```
## $`point estimate`
## [1] 2.208358
##
## $`error bound`
## [1] 0.022172
##
## $variance
## [1] 0.0001279716
```

5. Use step 4 to construct 95% confidence interval of population mean of “AGR1”. Repeat 100 times and calculate the relative frequency of those intervals containing the population mean.

```r
print(
  list(
    "CI.lower for mu" = ybar_syrs - error_bound,
    "CI.upper for mu" = ybar_syrs + error_bound
  )
)
```

```
## $`CI.lower for mu`
## [1] 2.186186
##
## $`CI.upper for mu`
## [1] 2.23053
```

```r
CI_cal <- function(data_mat, ns, n, N) {
  ybar <- rowMeans(sample_mat)
  ybar_syrs <- mean(ybar)
  s.2_ybar <- var(ybar)
  varhat_ybar_syrs <- (1 - n / N) * s.2_ybar / ns
  error_bound = qnorm(1 - alpha / 2) * sqrt(varhat_ybar_syrs)
  list(
    "CI.lower" = ybar_syrs - error_bound,
    "CI.upper" = ybar_syrs + error_bound,
    "ybarbar" = ybar_syrs
  )
}
mrep <- 1e2
temp <- logical(mrep)
for (i in 1:mrep) {
  start_vec = sample(1:k_prime, ns)
  sample_mat = matrix(NA, nrow = ns, ncol = n / ns)
  for (j in 1:ns) {
    sample_mat[j, ] = sy(data$AGR1, n / ns, k_prime, start_vec[j])
  }
  CI <- CI_cal(sample_mat, ns, n, N)
  temp[i] <- (CI[[1]] <= mu_agr1) * (mu_agr1 <= CI[[2]])
}
print(list(
  "relative frequency of those intervals containing the population mean" = mean(temp)
))
```

```
## $`relative frequency of those intervals containing the population mean`
## [1] 0.91
```