Sampling Practical Assignments

MSc Statistics Sem III

Omkar Ninav

Table of contents

Q.(1) The sugarcane production and cultivation of Indian states are presented in excel sheet Sugarcane Production cultivation. Draw a random sample using SRSWOR of size 10 and estimate the average amount of sugarcane production using the regression method of estimation and obtain the 95% confidence interval for the population mean.

```
library(dplyr)
library(tidyr)
set.seed(123)
sugarcane_data <- readxl::read_xlsx(
  "data/Sugarcane_Production_Cultivation.xlsx")


X <- sugarcane_data$`CultivatedArea (in lakh acres)`
Y <- sugarcane_data$`Production (in lakh tonnes)`
N <- length(Y)
n <- 10
s <- sample(1:N,n,replace = F)
x1 <- X[s]
x1
```

```
 [1] 1.83 0.17 0.05 2.43 0.98 0.30 1.14 9.02 1.84 0.30
```

```
y1 <- Y[s]
y1
```

```
 [1] 165.62  12.94   4.04 165.11  54.30  11.15  87.29 726.37 122.34  12.47
```

```r
m1 <- mean(X)
mx <- mean(x1)
my <- mean(y1)
sx_2 <- var(x1)
sx_2
```

```
[1] 7.094982
```

```r
sy_2 <- var(y1)
sy_2
```

```
[1] 46988.86
```

```r
s_xy <- cov(x1,y1)
beta_hat <- s_xy/sx_2
y_bar_reg <- my + beta_hat*(m1 - mx)

print(paste("Estimated mean using regression method is",y_bar_reg))
```

```
[1] "Estimated mean using regression method is 183.212596472875"
```

```r
#---95%CI for population mean---#
r <- cor(x1, y1)
MSE_reg <- (1/n - 1/N) * (sy_2 - r^2 * s_xy)

# Calculate t-critical value
alpha <- 0.05
df <- n - 2  # degrees of freedom for regression
t <- qt(1 - alpha/2, df)

# Confidence interval
LCL_reg <- y_bar_reg - t * sqrt(MSE_reg)
UCL_reg <- y_bar_reg + t * sqrt(MSE_reg)
CI_reg <- c(LCL_reg, UCL_reg)

cat("95% Confidence Interval: [", round(LCL_reg, 2), ",", round(UCL_reg, 2),
    "] lakh tonnes\n")
```

```
95% Confidence Interval: [ 72.12 , 294.3 ] lakh tonnes
```

Q.(2) Excel Sheet Maharashtra sex ratio gives India's census data of 2011, in which the number of females per thousand is measured district-wise and strata are regions. Draw a random sample by using SRSWOR of size 18 under the proportional allocation of stratified sampling. Estimate the average sex ratio and obtain the variance of the estimator.

```r
sex_ratio_data <- readxl::read_xlsx("data/maharashtra_sex_ratio_2011.xlsx")
colnames(sex_ratio_data) <- c("Sr_No", "Region", "District", "SexRatio")
N <- nrow(sex_ratio_data)
n <- 18
stratified_sample_data <- sex_ratio_data |>
  group_by(Region) |>
  sample_frac(size = n / N)
stratified_sample_data
```

```
# A tibble: 17 x 4
# Groups:   Region [6]
   Sr_No Region              District   SexRatio
   <dbl> <chr>               <chr>         <dbl>
 1    27 Khandesh            Dhule           941
 2    28 Khandesh            Jalgaon         922
 3    16 Konkan              Thane           880
 4    18 Konkan              Raigad          955
 5    20 Konkan              Sindhudurg     1037
 6    19 Konkan              Ratnagiri      1123
 7     6 Marathwada          Aurangabad      917
 8     9 Marathwada          Osmanabad       920
 9    13 Marathwada          Hingoli         935
10    11 Marathwada          Latur           924
11    36 Paschim Maharashtra Solapur         932
12    34 Paschim Maharashtra Sangli          964
13    22 Vidarbha            Chandrapur      959
14    26 Vidarbha            Wardha          946
15    21 Vidarbha            Bhandara        984
16     3 Vidarbha (Varhad)   Buldana         928
17     4 Vidarbha (Varhad)   Yavatmal        947
```

```r
population_summary <- sex_ratio_data |>
  group_by(Region) |>
  summarise(N_h = n()) |>
  mutate(N = sum(N_h),
         W_h = N_h / N)

sample_summary <- stratified_sample_data |>
  group_by(Region) |>
  summarise(n_h = n(),
            y_bar_h = mean(SexRatio),
            s_h2 = var(SexRatio))
stratified_summary <- merge(population_summary, sample_summary, by = "Region")
stratified_summary <- stratified_summary |>
  mutate(weighted_mean = W_h * y_bar_h,
         weighted_variance = (W_h^2) * (1 - n_h / N_h) * (s_h2 / n_h))
stratified_summary
```

```
               Region N_h  N       W_h n_h y_bar_h     s_h2 weighted_mean
1            Khandesh   5 36 0.1388889   2  931.50   180.50      129.3750
2              Konkan   7 36 0.1944444   4  998.75 10972.25      194.2014
3           Marathwada   8 36 0.2222222   4  924.00    62.00      205.3333
4 Paschim Maharashtra   5 36 0.1388889   2  948.00   512.00      131.6667
5            Vidarbha   6 36 0.1666667   3  963.00   373.00      160.5000
6   Vidarbha (Varhad)   5 36 0.1388889   2  937.50   180.50      130.2083
  weighted_variance
1          1.044560
2         44.447772
3          0.382716
4          2.962963
5          1.726852
6          1.044560
```

```
y_bar_strat <- sum(stratified_summary$weighted_mean)
var_y_bar_strat <- sum(stratified_summary$weighted_variance)
var_y_bar_strat
```

```
[1] 51.60942
```

Q.(3) Draw a random sample using SRSWOR of size 18 under optimal allocation of stratified sampling from Sex ratio data. Estimate the average sex ratio and obtain the variance of the estimator.

```
library(readxl)
library(dplyr)
library(tidyr)

# Load data
sex_ratio_data <- readxl::read_xlsx("data/maharashtra_sex_ratio_2011.xlsx")
colnames(sex_ratio_data) <- c("Sr_No", "Region", "District", "SexRatio")

# Total population size and sample size
N <- nrow(sex_ratio_data)
n <- 18

# Compute strata-level statistics
population_stats <- sex_ratio_data |>
  group_by(Region) |>
  summarise(
    N_h = n(),
    S_h = sd(SexRatio)
  ) |>
```

```r
  ungroup() |>
  replace_na(list(S_h = 0)) |>
  mutate(
    N = sum(N_h),
    W_h = N_h / N,
    Nh_Sh = N_h * S_h
  )

# Compute sum of Nh * Sh
sum_Nh_Sh <- sum(population_stats$Nh_Sh)

# Initial optimal n_h
sample_sizes_nh <- population_stats |>
  mutate(
    n_h_frac = n * (Nh_Sh / sum_Nh_Sh),
    n_h = floor(n_h_frac)
  )

# Each stratum gets at least 1
sample_sizes_nh$n_h <- ifelse(sample_sizes_nh$n_h < 1, 1, sample_sizes_nh$n_h)

# Ensure n_h does not exceed stratum population
sample_sizes_nh$n_h <- pmin(sample_sizes_nh$n_h, sample_sizes_nh$N_h)

# Adjust remaining to hit total n exactly
adjust_allocation <- function(df, target_n) {
  repeat {
    diff <- target_n - sum(df$n_h)
    if (diff == 0) break

    # allowable strata that can increase/decrease sample
    if (diff > 0) {
      eligible <- which(df$n_h < df$N_h)
      if (length(eligible) == 0) break
      add <- eligible[order(df$n_h_frac[eligible] - df$n_h[eligible],
                            decreasing = TRUE)][1:min(diff, length(eligible))]
      df$n_h[add] <- df$n_h[add] + 1
    } else {
      eligible <- which(df$n_h > 1)
      remove <- eligible[order(df$n_h_frac[eligible]
                               - df$n_h[eligible])][1:min(abs(diff),
                                                          length(eligible))]
      df$n_h[remove] <- df$n_h[remove] - 1
    }
  }
  df
}
```

```r
sample_sizes_nh <- adjust_allocation(sample_sizes_nh, n)

cat("\n Optimal allocation sample sizes (n_h) per Region:\n")
```

```
 Optimal allocation sample sizes (n_h) per Region:
```

```r
print(sample_sizes_nh |> select(Region, N_h, S_h, n_h))
```

```
# A tibble: 6 x 4
  Region                N_h   S_h   n_h
  <chr>               <int> <dbl> <dbl>
1 Khandesh                5  19.1     2
2 Konkan                  7 106.      7
3 Marathwada              8  10.1     2
4 Paschim Maharashtra     5  29.2     3
5 Vidarbha                6  20.3     2
6 Vidarbha (Varhad)       5  10.3     2
```

```r
cat("\n Final total sample size n =", sum(sample_sizes_nh$n_h), "\n")
```

```
 Final total sample size n = 18
```

```r
# --------------------------
# Draw Stratified Sample
# --------------------------
data_with_nh <- sex_ratio_data |>
  left_join(sample_sizes_nh |> select(Region, n_h), by = "Region")

stratified_sample_data_optimal <- data_with_nh |>
  group_by(Region) |>
  group_modify(~ slice_sample(.x, n = .x$n_h[1])) |>
  ungroup()

cat("\n Final Sample drawn: ", nrow(stratified_sample_data_optimal),
    " observations\n")
```

```
 Final Sample drawn:  18  observations
```

```r
# -------------------------
# Compute Weighted Stratified Mean
# -------------------------
sample_means_h <- stratified_sample_data_optimal |>
  group_by(Region) |>
  summarise(y_bar_h = mean(SexRatio), .groups = "drop")

est_mean_h <- population_stats |>
  select(Region, W_h) |>
  left_join(sample_means_h, by = "Region") |>
  mutate(weighted_mean = W_h * y_bar_h)

y_bar_strat_optimal <- sum(est_mean_h$weighted_mean)

cat("\n Estimated Stratified Mean (Optimal Allocation):",
    y_bar_strat_optimal, "\n")
```

```
 Estimated Stratified Mean (Optimal Allocation): 944.2824
```

```r
sample_vars_h <- stratified_sample_data_optimal |>
  group_by(Region) |>
  summarise(s_h2 = var(SexRatio)) |>
  replace_na(list(s_h2 = 0))

var_data <- population_stats |>
  select(Region, W_h, N_h) |>
  left_join(sample_sizes_nh |> select(Region, n_h), by = "Region")  |>
  left_join(sample_vars_h, by = "Region") |>
  replace_na(list(s_h2 = 0)) |>
  mutate(stratum_var_contrib = if_else(
      n_h > 0,  # Only calculate for strata we sampled from
      (W_h^2) * (1 - n_h / N_h) * (s_h2 / n_h),
      0         # Contribution is 0 if n_h = 0
    ))

est_var_strat_optimal <- sum(var_data$stratum_var_contrib)

print(paste("Variance of estimated mean (Optimal Allocation):",
            est_var_strat_optimal))
```

```
[1] "Variance of estimated mean (Optimal Allocation): 20.3923825445816"
```

Q.(4) Estimate the mean fuelwood_data consumption of the households of forest villages in the Dehradun district of Uttarakhand state and its standard error using the method of stratified random sampling. Three strata were constructed using the criterion of distance from the forest. Find

the estimated mean, total, and their variances. Data are shown excel sheet fuelwood_data consumption

```
fuelwood_data <- readxl::read_xlsx("data/fuelwood_consumption.xlsx")
colnames(fuelwood_data) <- c("Stratum","Selected_Unit", "Consumption")
# Population size
N <- nrow(fuelwood_data)

# Stratified Statistics
stats <- fuelwood_data|>
  group_by(Stratum)|>
  summarise(
    N_h = n(),
    ybar_h = mean(Consumption),
    S2_h = var(Consumption)
  )|>
  mutate(W_h = N_h / N)

stats
```

```
# A tibble: 3 x 5
  Stratum   N_h ybar_h  S2_h   W_h
  <chr>   <int>  <dbl> <dbl> <dbl>
1 I          10   4.22 1.05  0.435
2 II          5   4.42 0.335 0.217
3 III         8   5.46 2.00  0.348
```

```
# Estimated Stratified Mean
ybar_st <- sum(stats$W_h * stats$ybar_h)
ybar_st
```

```
[1] 4.692174
```

```
# Variance of Stratified Mean
var_ybar_st <- sum((stats$W_h^2 * stats$S2_h) / stats$N_h)
var_ybar_st
```

```
[1] 0.05323138
```

```
SE_ybar_st <- sqrt(var_ybar_st)
SE_ybar_st
```

```
[1] 0.2307193
```

```r
# Estimated Total fuelwood_data Consumption
T_st <- N * ybar_st
T_st
```

```
[1] 107.92
```

```r
# Variance of Total
var_Tst <- (N^2) * var_ybar_st
var_Tst
```

```
[1] 28.1594
```

Estimated Mean Consumption: 4.6922
Variance of Mean Estimator: 0.053231
Standard Error (Mean): 0.230719

Estimated Total Consumption: 107.92
Variance of Total Estimator: 28.1594

Q.(5) ICFRE has implemented a tree improvement programme of Gmelina arborea and has planned to release high productive pest-tolerant clones. Compute the sample size in each stratum under proportional and Neyman allocation. Calculate the sampling variance of the survived trees from the sample, if the plots were selected under

(i) proportional allocation and SRSWOR,
(ii) Neyman allocation and SRSWOR.

Data are shown in excel sheet Survival of clones

```r
survival_data <- readxl::read_xlsx("data/Survival of clones.xlsx")
colnames(survival_data) <- c("Stratum", "N_h","mean_h", "Sd_h")
# Total population plots
N <- sum(survival_data$N_h)

# Decide sample size
n <- 100  # <-- change here if a different sample size is specified

# Weight for each stratum
survival_data <- survival_data|>mutate(W_h = N_h / N)

# Proportional Allocation
survival_data <- survival_data|>
  mutate(n_prop = round(n * W_h))

# Neyman Allocation
sum_Nh_Sh <- sum(survival_data$N_h * survival_data$Sd_h)
survival_data <- survival_data|>
```

```
  mutate(n_neyman = round(n * (N_h * Sd_h) / sum_Nh_Sh))

survival_data
```

```
# A tibble: 5 x 7
  Stratum   N_h mean_h  Sd_h   W_h n_prop n_neyman
  <chr>   <dbl>  <dbl> <dbl> <dbl>  <dbl>    <dbl>
1 I         200    5.4   2.5 0.118     12        3
2 II        300   16.5  15.5 0.176     18       27
3 III       250   25.3  20.5 0.147     15       30
4 IV        430   15.6  12.5 0.253     25       31
5 V         520    7.5   3   0.306     31        9
```

```
# Check totals:
sum_prop <- sum(survival_data$n_prop)
sum_neyman <- sum(survival_data$n_neyman)

cat("\nTotal sample size under proportional allocation:", sum_prop)
```

```
Total sample size under proportional allocation: 101
```

```
cat("\nTotal sample size under Neyman allocation:", sum_neyman)
```

```
Total sample size under Neyman allocation: 100
```

```
# Proportional Allocation Variance
var_prop <- sum((survival_data$W_h^2 * survival_data$Sd_h^2)
                / survival_data$n_prop * (1 - survival_data$n_prop
                                          / survival_data$N_h))

# Neyman Allocation Variance
var_neyman <- sum((survival_data$W_h^2 * survival_data$Sd_h^2)
                  / survival_data$n_neyman * (1 - survival_data$n_neyman
                                              / survival_data$N_h))
```

0.0.1 Stratified Sampling Variance Results

Variance under proportional allocation: 1.369204
Standard Error (Proportional): 1.17013

Variance under Neyman allocation: 0.938335
Standard Error (Neyman): 0.968677

Q.(6) The experiment deals with the bamboo species Melocanna bamboosides with vessel lengths: 27, 38, 53, 43, 32, 45, 25, 32, 43, 22, 38, 42, 39, 34, 25, 27, 33, 22, 34, 48, 41, 34, 23, 37, 32, 37, 44, 41, 23, 41, 20, 29, 28, 39, 32, 27, 22, 37, 23, 32, 27, 23, 31, 26, 35, 43, 26, 24, 34, 22, 27, 30, 19, 12, 11, 14, 24, 25, 27, 20. Draw a systematic sample of size 6 and estimate the average vessel length.

```r
# Population data: vessel lengths
vessel_lengths <- c(27, 38, 53, 43, 32, 45, 25, 32, 43, 22, 38, 42, 39, 34, 25,
                    27, 33, 22, 34, 48, 41, 34, 23, 37, 32, 37, 44, 41, 23, 41,
                    20, 29, 28, 39, 32, 27, 22, 37, 23, 32, 27, 23, 31, 26, 35,
                    43, 26, 24, 34, 22, 27, 30, 19, 12, 11, 14, 24, 25, 27, 20)

# Population size
N <- length(vessel_lengths)

# Sample size
n <- 6

# Calculate sampling interval
k <- floor(N / n)

cat("Population size (N):", N, "\n")
```

```
Population size (N): 60
```

```r
cat("Sample size (n):", n, "\n")
```

```
Sample size (n): 6
```

```r
cat("Sampling interval (k):", k, "\n\n")
```

```
Sampling interval (k): 10
```

```r
# Select random start between 1 and k
set.seed(123)  # For reproducibility
r <- sample(1:k, size = 1)

cat("Random start (r):", r, "\n\n")
```

```
Random start (r): 3
```

```r
# Draw systematic sample
sample_indices <- r + (0:(n - 1)) * k
systematic_sample <- vessel_lengths[sample_indices]

cat("Selected indices:", sample_indices, "\n")
```

```
Selected indices: 3 13 23 33 43 53
```

```
cat("Systematic sample:", systematic_sample, "\n\n")
```

```
Systematic sample: 53 39 23 28 31 19
```

```
# Estimate average vessel length
y_bar_sys <- mean(systematic_sample)

cat(strrep("=", 60), "\n")
```

```
============================================================
```

```
cat("SYSTEMATIC SAMPLING RESULTS\n")
```

```
SYSTEMATIC SAMPLING RESULTS
```

```
cat(strrep("=", 60), "\n")
```

```
============================================================
```

```
cat("Estimated average vessel length (ȳ_sys):", round(y_bar_sys, 2), "\n")
```

```
Estimated average vessel length (ȳ_sys): 32.17
```

```
cat("Population mean (for comparison):", round(mean(vessel_lengths), 2), "\n")
```

```
Population mean (for comparison): 30.73
```

Explanation:

Systematic Sampling Formula:

- Sampling interval: $k = \lfloor N/n \rfloor = \lfloor 60/6 \rfloor = 10$
- Random start: $r \sim \text{Uniform}(1, 2, \ldots, k)$
- Selected units: $r, r+k, r+2k, \ldots, r+(n-1)k$

Estimator:

$$\bar{y}_{sys} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

The systematic sample mean provides an unbiased estimate of the population mean under the assumption that there is no periodic pattern in the population.

Q.(7) Water samples collected from different locations of Pune district are clustered in excel sheet Water Quality Pune. Draw a random sample of 3 clusters and obtain the average conductivity (ppm) of the water and its variance.

```
# Load water quality data
water_data <- readxl::read_xlsx("data/Water quality Pune.xlsx")

# Display structure of data
cat("Data structure:\n")
```

```
Data structure:
```

```
print(head(water_data))
```

```
# A tibble: 6 x 9
  `Sr. No.` Cluster   Site    PH `Conductivity (ppm)` Chloride Sulfate
      <dbl> <chr>    <dbl> <dbl>                <dbl>    <dbl>   <dbl>
1         1 Baramati     1  7.4                   730      332     157
2         2 Baramati     2  8.21                  280      113      27
3         3 Baramati     3  7.48                  460      119     838
4         4 Baramati     4  7.62                  450      142      71
5         5 Baramati     5  7.97                  400      151     192
6         6 Baramati     6  7.96                  480      156     192
# i 2 more variables: Phosphorous <dbl>, Nitrate <dbl>
```

```
cat("\n")
```

```
# Get unique clusters
clusters <- unique(water_data$Cluster)
M <- length(clusters)  # Total number of clusters
m <- 3  # Number of clusters to sample

cat("Total number of clusters (M):", M, "\n")
```

```
Total number of clusters (M): 6
```

```r
cat("Number of clusters to sample (m):", m, "\n\n")
```

```
Number of clusters to sample (m): 3
```

```r
# Draw random sample of 3 clusters using SRSWOR
set.seed(123)  # For reproducibility
sampled_clusters <- sample(clusters, size = m, replace = FALSE)

cat("Sampled clusters:", sampled_clusters, "\n\n")
```

```
Sampled clusters: Purandar Indapur Daund
```

```r
# Extract data from sampled clusters
sample_data <- water_data[water_data$Cluster %in% sampled_clusters, ]

cat("Sample size:", nrow(sample_data), "observations\n\n")
```

```
Sample size: 18 observations
```

```r
# Calculate cluster means
cluster_means <- tapply(sample_data$`Conductivity (ppm)`,
                        sample_data$Cluster,
                        mean)

cat("Cluster means (ȳ_i):\n")
```

```
Cluster means (ȳ_i):
```

```r
print(round(cluster_means, 2))
```

```
   Daund  Indapur Purandar
  615.00   683.33   305.00
```

```r
cat("\n")
```

```r
# One-stage cluster sampling estimator
y_bar_cluster <- mean(cluster_means)

cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```r
cat("CLUSTER SAMPLING RESULTS\n")
```

```
CLUSTER SAMPLING RESULTS
```

```r
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```r
cat("Average conductivity estimate (ȳ_cl):", round(y_bar_cluster, 2), "ppm\n\n")
```

```
Average conductivity estimate (ȳ_cl): 534.44 ppm
```

```r
# Calculate variance of cluster mean estimator
# Between-cluster variance
s_b_squared <- var(cluster_means)

# Finite Population Correction
fpc <- (M - m) / M

# Variance of cluster mean
var_y_cluster <- fpc * s_b_squared / m

# Standard error
se_y_cluster <- sqrt(var_y_cluster)

cat("Variance Components:\n")
```

```
Variance Components:
```

```r
cat("  Between-cluster variance (s²_b):", round(s_b_squared, 4), "\n")
```

```
  Between-cluster variance (s²_b): 40650.93
```

```r
cat("  FPC factor (1 - m/M):", round(fpc, 4), "\n")
```

```
  FPC factor (1 - m/M): 0.5
```

```r
cat("  Number of sampled clusters (m):", m, "\n\n")
```

```
  Number of sampled clusters (m): 3
```

```r
cat("Variance of cluster mean estimator:", round(var_y_cluster, 4), "\n")
```

```
Variance of cluster mean estimator: 6775.154
```

```r
cat("Standard error:", round(se_y_cluster, 4), "ppm\n\n")
```

```
Standard error: 82.3113 ppm
```

```r
# 95% Confidence Interval
t_critical <- qt(0.975, df = m - 1)
ci_lower <- y_bar_cluster - t_critical * se_y_cluster
ci_upper <- y_bar_cluster + t_critical * se_y_cluster

cat("95% Confidence Interval: [", round(ci_lower, 2), ",",
    round(ci_upper, 2), "] ppm\n")
```

```
95% Confidence Interval: [ 180.29 , 888.6 ] ppm
```

Explanation:

One-Stage Cluster Sampling Formulas:

Cluster Mean Estimator:

$$\bar{y}_{cl} = \frac{1}{m}\sum_{i=1}^{m}\bar{y}_i$$

where $\bar{y}_i$ is the mean conductivity in cluster i.

Variance of Cluster Mean:

$$V(\bar{y}_{cl}) = \left(1 - \frac{m}{M}\right)\frac{s_b^2}{m}$$

where:

- M = total number of clusters in population
- m = number of clusters sampled
- s_b^2 = between-cluster variance = $\frac{1}{m-1}\sum_{i=1}^{m}(\bar{y}_i - \bar{y}_{cl})^2$
- $(1 - \frac{m}{M})$ = finite population correction factor

Key Points:

- Cluster sampling is efficient when clusters are heterogeneous (different from each other) but internally homogeneous
- The variance depends on between-cluster variability
- More homogeneous clusters (similar means) → lower variance

- More heterogeneous clusters (different means) → higher variance

Q.(8) Draw a random sample of 3 clusters and then draw a random sample of size 3 from each of the selected clusters. Obtain the average conductivity of water and its variance. Use the R code.

```
# TWO-STAGE CLUSTER SAMPLING
# Load water quality data
water_data <- readxl::read_xlsx("data/Water quality Pune.xlsx")

# Get unique clusters
clusters <- unique(water_data$Cluster)
M <- length(clusters)  # Total number of clusters
m <- 3  # Number of clusters to sample (first stage)
n_per_cluster <- 3  # Sample size from each cluster (second stage)

cat("TWO-STAGE CLUSTER SAMPLING\n")
```

```
TWO-STAGE CLUSTER SAMPLING
```

```
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```
cat("Total number of clusters (M):", M, "\n")
```

```
Total number of clusters (M): 6
```

```
cat("Clusters to sample (m):", m, "\n")
```

```
Clusters to sample (m): 3
```

```
cat("Units per cluster (n_i):", n_per_cluster, "\n\n")
```

```
Units per cluster (n_i): 3
```

```
# STAGE 1: Select 3 clusters using SRSWOR
set.seed(456)  # For reproducibility
sampled_clusters <- sample(clusters, size = m, replace = FALSE)

cat("STAGE 1: Selected clusters:", sampled_clusters, "\n\n")
```

```
STAGE 1: Selected clusters: Khed Indapur Purandar
```

```r
# Get cluster sizes
cluster_sizes <- table(water_data$Cluster)
N_i <- cluster_sizes[as.character(sampled_clusters)]

cat("Cluster sizes (N_i):\n")
```

```
Cluster sizes (N_i):
```

```r
print(N_i)
```

```
    Khed  Indapur Purandar
       6        6        6
```

```r
cat("\n")
```

```r
# STAGE 2: Sample n_per_cluster units from each selected cluster
two_stage_sample <- NULL
cluster_sample_means <- numeric(m)
cluster_sample_vars <- numeric(m)

cat("STAGE 2: Sampling from each cluster\n")
```

```
STAGE 2: Sampling from each cluster
```

```r
cat(strrep("-", 70), "\n")
```

```
----------------------------------------------------------------------
```

```r
for (i in 1:m) {
  cluster_id <- sampled_clusters[i]

  # Get all units from this cluster
  cluster_data <- water_data[water_data$Cluster == cluster_id, ]

  # Sample n_per_cluster units from this cluster
  sampled_indices <- sample(1:nrow(cluster_data),
                            size = n_per_cluster,
                            replace = FALSE)
  cluster_subsample <- cluster_data[sampled_indices, ]

  # Calculate cluster mean and variance from subsample
  cluster_sample_means[i] <- mean(cluster_subsample$`Conductivity (ppm)`)
```

```
  cluster_sample_vars[i] <- var(cluster_subsample$`Conductivity (ppm)`)

  # Store subsample
  two_stage_sample <- rbind(two_stage_sample, cluster_subsample)

  cat("Cluster", cluster_id, ":\n")
  cat("  Sampled values:",
      round(cluster_subsample$`Conductivity (ppm)`, 2), "\n")
  cat("  Subsample mean (ȳ_i):", round(cluster_sample_means[i], 2), "\n")
  cat("  Subsample variance (s²_i):", round(cluster_sample_vars[i], 4), "\n\n")
}
```

```
Cluster Khed :
  Sampled values: 430 780 480
  Subsample mean (ȳ_i): 563.33
  Subsample variance (s²_i): 35833.33

Cluster Indapur :
  Sampled values: 880 1040 510
  Subsample mean (ȳ_i): 810
  Subsample variance (s²_i): 73900

Cluster Purandar :
  Sampled values: 550 70 180
  Subsample mean (ȳ_i): 266.67
  Subsample variance (s²_i): 63233.33
```

```
# Two-stage estimator: mean of cluster means
y_bar_2stage <- mean(cluster_sample_means)

cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```
cat("TWO-STAGE CLUSTER SAMPLING RESULTS\n")
```

```
TWO-STAGE CLUSTER SAMPLING RESULTS
```

```
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```r
cat("Average conductivity estimate (ȳ_2s):", round(y_bar_2stage, 2), "ppm\n\n")
```

```
Average conductivity estimate (ȳ_2s): 546.67 ppm
```

```r
# VARIANCE CALCULATION for Two-Stage Cluster Sampling
# Component 1: Between-cluster variance
s_b_squared_2s <- var(cluster_sample_means)
fpc_clusters <- (M - m) / M
var_between <- fpc_clusters * s_b_squared_2s / m

# Component 2: Within-cluster variance
# Average of within-cluster variances weighted by sampling fractions
var_within <- 0
for (i in 1:m) {
  fpc_within <- (N_i[i] - n_per_cluster) / N_i[i]
  var_within <- var_within + fpc_within * cluster_sample_vars[i] / n_per_cluster
}
var_within <- var_within / m

# Total variance
var_y_2stage <- var_between + var_within
se_y_2stage <- sqrt(var_y_2stage)

cat("Variance Components:\n")
```

```
Variance Components:
```

```r
cat("  Between-cluster variance component:", round(var_between, 4), "\n")
```

```
  Between-cluster variance component: 12335.19
```

```r
cat("  Within-cluster variance component:", round(var_within, 4), "\n")
```

```
  Within-cluster variance component: 9609.259
```

```r
cat("  Total variance V(ȳ_2s):", round(var_y_2stage, 4), "\n")
```

```
  Total variance V(ȳ_2s): 21944.44
```

```r
cat("  Standard error:", round(se_y_2stage, 4), "ppm\n\n")
```

```
  Standard error: 148.1366 ppm
```

```r
# 95% Confidence Interval
# Use t-distribution with m-1 degrees of freedom (conservative)
t_crit_2s <- qt(0.975, df = m - 1)
ci_lower_2s <- y_bar_2stage - t_crit_2s * se_y_2stage
ci_upper_2s <- y_bar_2stage + t_crit_2s * se_y_2stage

cat("95% Confidence Interval: [", round(ci_lower_2s, 2), ",",
    round(ci_upper_2s, 2), "] ppm\n\n")
```

```
95% Confidence Interval: [ -90.71 , 1184.05 ] ppm
```

```r
# Summary table
summary_table <- data.frame(
  Cluster = sampled_clusters,
  N_i = as.numeric(N_i),
  n_i = rep(n_per_cluster, m),
  y_bar_i = round(cluster_sample_means, 2),
  s2_i = round(cluster_sample_vars, 4)
)

cat("Summary Table:\n")
```

```
Summary Table:
```

```r
print(summary_table)
```

```
   Cluster N_i n_i y_bar_i     s2_i
1     Khed   6   3  563.33 35833.33
2  Indapur   6   3  810.00 73900.00
3 Purandar   6   3  266.67 63233.33
```

Explanation:

Two-Stage Cluster Sampling Formulas:

Stage 1: Select m clusters from M total clusters using SRSWOR
Stage 2: From each selected cluster i, select n_i units from N_i total units using SRSWOR

Two-Stage Mean Estimator:

$$\bar{y}_{2s} = \frac{1}{m} \sum_{i=1}^{m} \bar{y}_i$$

where $\bar{y}_i$ is the mean of the subsample from cluster i.

Variance of Two-Stage Mean:

$$V(\bar{y}_{2s}) = \underbrace{\left(1 - \frac{m}{M}\right)\frac{s_b^2}{m}}_{\text{Between-cluster}} + \underbrace{\frac{1}{m}\sum_{i=1}^{m}\left(1 - \frac{n_i}{N_i}\right)\frac{s_i^2}{n_i}}_{\text{Within-cluster}}$$

Components:

1. **Between-cluster variance** - Variability due to selecting only m out of M clusters
2. **Within-cluster variance** - Variability due to sampling only n_i out of N_i units from each cluster

Key Differences from One-Stage:

- One-stage: All units in selected clusters are measured
- Two-stage: Only a subsample of units from selected clusters is measured
- Two-stage has additional variance from the second-stage sampling
- Two-stage is more cost-effective when measuring all units is expensive

Q.(9) Select a first-phase sample of 20 units by SRSWOR sampling and note only the age of a person in a block from the selected units present in excel sheet sleeping hours. From the selected first-phase sample of 20 units, select a subsample of 10 units and note age and sleeping hours. Estimate the average sleeping hours using a ratio estimator in two-phase sampling. Deduce the 95% confidence interval.

```
# Load sleeping hours data
sleep_data <- readxl::read_xlsx("data/Sleeping Hours.xlsx")

# Variables
y <- sleep_data$`Number of hours of sleep`  # Variable of interest (expensive)
x <- sleep_data$Age                          # Auxiliary variable (cheap)

N <- length(y)  # Population size
n1 <- 20        # First-phase sample size
n2 <- 10        # Second-phase sample size

cat("TWO-PHASE RATIO ESTIMATION\n")
```

```
TWO-PHASE RATIO ESTIMATION
```

```
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```
cat("Population size (N):", N, "\n")
```

```
Population size (N): 50
```

```r
cat("First-phase sample size (n'):", n1, "\n")
```

```
First-phase sample size (n'): 20
```

```r
cat("Second-phase sample size (n):", n2, "\n\n")
```

```
Second-phase sample size (n): 10
```

```r
# PHASE 1: Select 20 units by SRSWOR and measure only Age (x)
set.seed(789)
phase1_indices <- sample(1:N, n1, replace = FALSE)

# First-phase data (only x is measured)
x_phase1 <- x[phase1_indices]

cat("PHASE 1: Measure auxiliary variable (Age) only\n")
```

```
PHASE 1: Measure auxiliary variable (Age) only
```

```r
cat(strrep("-", 70), "\n")
```

```
----------------------------------------------------------------------
```

```r
cat("Selected indices:", phase1_indices, "\n")
```

```
Selected indices: 45 48 12 42 26 35 37 36 3 6 41 29 21 49 2 24 50 30 39 27
```

```r
cat("Ages measured:", x_phase1, "\n\n")
```

```
Ages measured: 70.84 60.06 64.94 76.66 64.06 63.06 68.55 71.14 64.49 58.91 64.89 76.42 67.1 68
```

```r
# Calculate first-phase mean of x
x_bar_prime <- mean(x_phase1)

cat("First-phase mean of Age (x̄'):", round(x_bar_prime, 2), "years\n\n")
```

```
First-phase mean of Age (x̄'): 66.13 years
```

```r
# PHASE 2: Select 10 units from phase 1 sample and measure both Age and Sleep
set.seed(790)
phase2_indices_within <- sample(1:n1, n2, replace = FALSE)
phase2_indices <- phase1_indices[phase2_indices_within]

# Second-phase data (both x and y measured)
x_phase2 <- x[phase2_indices]
y_phase2 <- y[phase2_indices]

cat("PHASE 2: Measure both Age and Sleeping Hours\n")
```

```
PHASE 2: Measure both Age and Sleeping Hours
```

```r
cat(strrep("-", 70), "\n")
```

```
----------------------------------------------------------------------
```

```r
cat("Selected indices from phase 1:", phase2_indices_within, "\n")
```

```
Selected indices from phase 1: 8 3 9 11 17 19 2 15 6 20
```

```r
cat("Actual indices in population:", phase2_indices, "\n")
```

```
Actual indices in population: 36 12 3 41 50 39 48 2 35 27
```

```r
cat("Ages:", x_phase2, "\n")
```

```
Ages: 71.14 64.94 64.49 64.89 58.66 51.64 60.06 73.82 63.06 70.33
```

```r
cat("Sleep hours:", round(y_phase2, 2), "\n\n")
```

```
Sleep hours: 326.33 401.95 395.99 419.98 412.7 434.5 425.12 367.02 381.98 332.56
```

```r
# Calculate second-phase means
x_bar <- mean(x_phase2)
y_bar <- mean(y_phase2)

cat("Second-phase mean of Age (x̄):", round(x_bar, 2), "years\n")
```

```
Second-phase mean of Age (x̄): 64.3 years
```

```r
cat("Second-phase mean of Sleep (ȳ):", round(y_bar, 2), "hours\n\n")
```

```
Second-phase mean of Sleep (ȳ): 389.81 hours
```

```r
# TWO-PHASE RATIO ESTIMATOR
R_hat <- y_bar / x_bar
y_bar_rd <- R_hat * x_bar_prime

cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```r
cat("RATIO ESTIMATOR RESULTS\n")
```

```
RATIO ESTIMATOR RESULTS
```

```r
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```r
cat("Sample ratio R = ȳ/x̄:", round(R_hat, 4), "\n")
```

```
Sample ratio R = ȳ/x̄: 6.0621
```

```r
cat("Two-phase ratio estimate (ȳ_rd):", round(y_bar_rd, 4), "hours\n\n")
```

```
Two-phase ratio estimate (ȳ_rd): 400.8643 hours
```

```r
# VARIANCE CALCULATION
# Calculate variances and covariance from phase 2
s_y_sq <- var(y_phase2)
s_x_sq <- var(x_phase2)
s_xy <- cov(x_phase2, y_phase2)

# Variance of x from phase 1
s_x_prime_sq <- var(x_phase1)

# Two-phase ratio variance formula
# V(ȳ_rd)  (1/n - 1/n')[s²_y + R²s²_x - 2Rs_xy] + R²s²_x'/n'
term1 <- (1/n2 - 1/n1) * (s_y_sq + R_hat^2 * s_x_sq - 2 * R_hat * s_xy)
term2 <- (R_hat^2 * s_x_prime_sq) / n1
```

```r
var_y_rd <- term1 + term2
se_y_rd <- sqrt(var_y_rd)

cat("Variance Components:\n")
```

```
Variance Components:
```

```r
cat("  s²_y (sleep variance, phase 2):", round(s_y_sq, 4), "\n")
```

```
  s²_y (sleep variance, phase 2): 1416.857
```

```r
cat("  s²_x (age variance, phase 2):", round(s_x_sq, 4), "\n")
```

```
  s²_x (age variance, phase 2): 42.9079
```

```r
cat("  s_xy (covariance, phase 2):", round(s_xy, 4), "\n")
```

```
  s_xy (covariance, phase 2): -200.502
```

```r
cat("  s²_x' (age variance, phase 1):", round(s_x_prime_sq, 4), "\n\n")
```

```
  s²_x' (age variance, phase 1): 42.9368
```

```r
cat("  Term 1 (second-phase component):", round(term1, 6), "\n")
```

```
  Term 1 (second-phase component): 271.2316
```

```r
cat("  Term 2 (first-phase component):", round(term2, 6), "\n\n")
```

```
  Term 2 (first-phase component): 78.89515
```

```r
cat("Variance of ratio estimator V(ȳ_rd):", round(var_y_rd, 6), "\n")
```

```
Variance of ratio estimator V(ȳ_rd): 350.1268
```

```r
cat("Standard error SE(ȳ_rd):", round(se_y_rd, 4), "hours\n\n")
```

```
Standard error SE(ȳ_rd): 18.7117 hours
```

```r
# 95% CONFIDENCE INTERVAL
# Use z-distribution for large samples
z_critical <- qnorm(0.975)  # 1.96 for 95% CI

ci_lower_rd <- y_bar_rd - z_critical * se_y_rd
ci_upper_rd <- y_bar_rd + z_critical * se_y_rd

cat("95% CONFIDENCE INTERVAL\n")
```

```
95% CONFIDENCE INTERVAL
```

```r
cat(strrep("-", 70), "\n")
```

```
----------------------------------------------------------------------
```

```r
cat("Point estimate:", round(y_bar_rd, 4), "hours\n")
```

```
Point estimate: 400.8643 hours
```

```r
cat("Standard error:", round(se_y_rd, 4), "hours\n")
```

```
Standard error: 18.7117 hours
```

```r
cat("Critical value (z_0.025):", round(z_critical, 4), "\n")
```

```
Critical value (z_0.025): 1.96
```

```r
cat("95% CI: [", round(ci_lower_rd, 4), ",", round(ci_upper_rd, 4), "] hours\n\n")
```

```
95% CI: [ 364.19 , 437.5385 ] hours
```

```r
# Correlation
r_xy <- cor(x_phase2, y_phase2)
cat("Correlation between Age and Sleep (r):", round(r_xy, 4), "\n")
```

```
Correlation between Age and Sleep (r): -0.8132
```

```r
# Summary
cat("\n")
```

```
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```
cat("SUMMARY\n")
```

```
SUMMARY
```

```
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```
cat("Based on the two-phase ratio estimation:\n")
```

```
Based on the two-phase ratio estimation:
```

```
cat("  - Estimated average sleeping hours:", round(y_bar_rd, 2), "hours\n")
```

```
  - Estimated average sleeping hours: 400.86 hours
```

```
cat("  - We are 95% confident the true mean is between\n")
```

```
  - We are 95% confident the true mean is between
```

```
cat("    ", round(ci_lower_rd, 2), "and", round(ci_upper_rd, 2), "hours\n")
```

```
    364.19 and 437.54 hours
```

Explanation:

Two-Phase Sampling Design:

Two-phase (double) sampling is useful when: - Measuring auxiliary variable x (Age) is cheap - Measuring variable of interest y (Sleep) is expensive - x and y are correlated

Two-Phase Ratio Estimator:

$$\bar{y}_{rd} = \frac{\bar{y}}{\bar{x}} \times \bar{x}'$$

where: - $\bar{y}$ = mean of y from second-phase sample (size n) - $\bar{x}$ = mean of x from second-phase sample (size n) - $\bar{x}'$ = mean of x from first-phase sample (size n') - $\hat{R} = \bar{y}/\bar{x}$ = sample ratio

Variance Formula:

$$V(\bar{y}_{rd}) \approx \left(\frac{1}{n} - \frac{1}{n'}\right)[s_y^2 + \hat{R}^2 s_x^2 - 2\hat{R}s_{xy}] + \frac{\hat{R}^2 s_{x'}^2}{n'}$$

Components:

1. **First term** - Variance from ratio estimation in second phase
2. **Second term** - Variance from estimating $\bar{X}$ using first phase

Advantages:

- More cost-effective than measuring y on all n' units
- Uses correlation between x and y to improve precision
- $\bar{x}'$ is more precise than $\bar{x}$ (larger sample size)

Q.(10) Repeat the above procedure using a regression estimator in two-phase sampling. Deduce the 95% confidence interval.

```
# Load sleeping hours data (use same data as Q.9)
sleep_data <- readxl::read_xlsx("data/Sleeping Hours.xlsx")

# Variables
y <- sleep_data$`Number of hours of sleep`  # Variable of interest (expensive)
x <- sleep_data$Age                          # Auxiliary variable (cheap)

N <- length(y)  # Population size
n1 <- 20        # First-phase sample size
n2 <- 10        # Second-phase sample size

cat("TWO-PHASE REGRESSION ESTIMATION\n")
```

```
TWO-PHASE REGRESSION ESTIMATION
```

```
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```
cat("Population size (N):", N, "\n")
```

```
Population size (N): 50
```

```
cat("First-phase sample size (n'):", n1, "\n")
```

```
First-phase sample size (n'): 20
```

```r
cat("Second-phase sample size (n):", n2, "\n\n")
```

```
Second-phase sample size (n): 10
```

```r
# PHASE 1: Select 20 units by SRSWOR and measure only Age (x)
set.seed(789)  # Same seed as Q.9 for comparison
phase1_indices <- sample(1:N, n1, replace = FALSE)

# First-phase data (only x is measured)
x_phase1 <- x[phase1_indices]

cat("PHASE 1: Measure auxiliary variable (Age) only\n")
```

```
PHASE 1: Measure auxiliary variable (Age) only
```

```r
cat(strrep("-", 70), "\n")
```

```
----------------------------------------------------------------------
```

```r
cat("Selected indices:", phase1_indices, "\n\n")
```

```
Selected indices: 45 48 12 42 26 35 37 36 3 6 41 29 21 49 2 24 50 30 39 27
```

```r
# Calculate first-phase mean of x
x_bar_prime <- mean(x_phase1)

cat("First-phase mean of Age (x̄'):", round(x_bar_prime, 2), "years\n\n")
```

```
First-phase mean of Age (x̄'): 66.13 years
```

```r
# PHASE 2: Select 10 units from phase 1 sample and measure both Age and Sleep
set.seed(790)  # Same seed as Q.9 for comparison
phase2_indices_within <- sample(1:n1, n2, replace = FALSE)
phase2_indices <- phase1_indices[phase2_indices_within]

# Second-phase data (both x and y measured)
x_phase2 <- x[phase2_indices]
y_phase2 <- y[phase2_indices]

cat("PHASE 2: Measure both Age and Sleeping Hours\n")
```

```
PHASE 2: Measure both Age and Sleeping Hours
```

```r
cat(strrep("-", 70), "\n")
```

```
----------------------------------------------------------------------
```

```r
cat("Selected indices from phase 1:", phase2_indices_within, "\n")
```

```
Selected indices from phase 1: 8 3 9 11 17 19 2 15 6 20
```

```r
cat("Ages:", x_phase2, "\n")
```

```
Ages: 71.14 64.94 64.49 64.89 58.66 51.64 60.06 73.82 63.06 70.33
```

```r
cat("Sleep hours:", round(y_phase2, 2), "\n\n")
```

```
Sleep hours: 326.33 401.95 395.99 419.98 412.7 434.5 425.12 367.02 381.98 332.56
```

```r
# Calculate second-phase means
x_bar <- mean(x_phase2)
y_bar <- mean(y_phase2)

cat("Second-phase mean of Age (x̄):", round(x_bar, 2), "years\n")
```

```
Second-phase mean of Age (x̄): 64.3 years
```

```r
cat("Second-phase mean of Sleep (ȳ):", round(y_bar, 2), "hours\n\n")
```

```
Second-phase mean of Sleep (ȳ): 389.81 hours
```

```r
# TWO-PHASE REGRESSION ESTIMATOR
# Fit regression model using phase 2 data
model <- lm(y_phase2 ~ x_phase2)
beta_hat <- coef(model)[2]  # Regression slope
alpha_hat <- coef(model)[1] # Intercept

# Regression estimator
y_bar_ld <- y_bar + beta_hat * (x_bar_prime - x_bar)

cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```r
cat("REGRESSION ESTIMATOR RESULTS\n")
```

```
REGRESSION ESTIMATOR RESULTS
```

```r
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```r
cat("Regression model: Sleep =   +   × Age\n")
```

```
Regression model: Sleep =   +   × Age
```

```r
cat("  Intercept (̂):", round(alpha_hat, 4), "\n")
```

```
  Intercept (̂): 690.291
```

```r
cat("  Slope (̂):", round(beta_hat, 4), "\n")
```

```
  Slope (̂): -4.6728
```

```r
cat("  Adjustment (x̄' - x̄):", round(x_bar_prime - x_bar, 4), "\n\n")
```

```
  Adjustment (x̄' - x̄): 1.823
```

```r
cat("Two-phase regression estimate (ȳ_ld):", round(y_bar_ld, 4), "hours\n\n")
```

```
Two-phase regression estimate (ȳ_ld): 381.2944 hours
```

```r
# VARIANCE CALCULATION
# Method 1: Using correlation
s_y_sq <- var(y_phase2)
s_x_sq <- var(x_phase2)
r_xy <- cor(x_phase2, y_phase2)

# Variance of x from phase 1
s_x_prime_sq <- var(x_phase1)

# Two-phase regression variance formula
# V(ȳ_ld)  (1/n - 1/n')s²_y(1 - r²) + s²_x'/n'
var_y_ld_method1 <- (1/n2 - 1/n1) * s_y_sq * (1 - r_xy^2) + s_x_prime_sq / n1
```

```r
# Method 2: Using MSE from regression
residuals <- resid(model)
MSE <- sum(residuals^2) / (n2 - 2)
var_y_ld_method2 <- (1/n2 - 1/n1) * MSE + s_x_prime_sq / n1

# Use Method 1 for standard error
se_y_ld <- sqrt(var_y_ld_method1)

cat("Variance Components:\n")
```

```
Variance Components:
```

```r
cat("  s²_y (sleep variance, phase 2):", round(s_y_sq, 4), "\n")
```

```
  s²_y (sleep variance, phase 2): 1416.857
```

```r
cat("  s²_x (age variance, phase 2):", round(s_x_sq, 4), "\n")
```

```
  s²_x (age variance, phase 2): 42.9079
```

```r
cat("  s²_x' (age variance, phase 1):", round(s_x_prime_sq, 4), "\n")
```

```
  s²_x' (age variance, phase 1): 42.9368
```

```r
cat("  Correlation r_xy:", round(r_xy, 4), "\n")
```

```
  Correlation r_xy: -0.8132
```

```r
cat("  R² (variance explained):", round(r_xy^2, 4), "\n")
```

```
  R² (variance explained): 0.6613
```

```r
cat("  MSE (residual variance):", round(MSE, 4), "\n\n")
```

```
  MSE (residual variance): 539.9347
```

```r
cat("Variance Calculation:\n")
```

```
Variance Calculation:
```

```
cat("  Method 1 (using r²):", round(var_y_ld_method1, 6), "\n")
```

```
  Method 1 (using r²): 26.14394
```

```
cat("  Method 2 (using MSE):", round(var_y_ld_method2, 6), "\n\n")
```

```
  Method 2 (using MSE): 29.14358
```

```
cat("Variance of regression estimator V(ȳ_ld):", round(var_y_ld_method1, 6), "\n")
```

```
Variance of regression estimator V(ȳ_ld): 26.14394
```

```
cat("Standard error SE(ȳ_ld):", round(se_y_ld, 4), "hours\n\n")
```

```
Standard error SE(ȳ_ld): 5.1131 hours
```

```
# 95% CONFIDENCE INTERVAL
# Use z-distribution for large samples
z_critical <- qnorm(0.975)  # 1.96 for 95% CI

ci_lower_ld <- y_bar_ld - z_critical * se_y_ld
ci_upper_ld <- y_bar_ld + z_critical * se_y_ld

cat("95% CONFIDENCE INTERVAL\n")
```

```
95% CONFIDENCE INTERVAL
```

```
cat(strrep("-", 70), "\n")
```

```
----------------------------------------------------------------------
```

```
cat("Point estimate:", round(y_bar_ld, 4), "hours\n")
```

```
Point estimate: 381.2944 hours
```

```
cat("Standard error:", round(se_y_ld, 4), "hours\n")
```

```
Standard error: 5.1131 hours
```

```r
cat("Critical value (z_0.025):", round(z_critical, 4), "\n")
```

```
Critical value (z_0.025): 1.96
```

```r
cat("95% CI: [", round(ci_lower_ld, 4), ",", round(ci_upper_ld, 4), "] hours\n\n")
```

```
95% CI: [ 371.2729 , 391.3159 ] hours
```

```r
# COMPARISON WITH RATIO ESTIMATOR (from Q.9)
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```r
cat("COMPARISON: RATIO vs REGRESSION ESTIMATORS\n")
```

```
COMPARISON: RATIO vs REGRESSION ESTIMATORS
```

```r
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```r
# Recalculate ratio estimator for comparison
R_hat <- y_bar / x_bar
y_bar_rd <- R_hat * x_bar_prime
s_xy <- cov(x_phase2, y_phase2)
var_y_rd <- (1/n2 - 1/n1) * (s_y_sq + R_hat^2 * s_x_sq - 2 * R_hat * s_xy) +
            (R_hat^2 * s_x_prime_sq) / n1
se_y_rd <- sqrt(var_y_rd)

comparison_table <- data.frame(
  Estimator = c("Ratio", "Regression"),
  Estimate = c(round(y_bar_rd, 4), round(y_bar_ld, 4)),
  Variance = c(round(var_y_rd, 6), round(var_y_ld_method1, 6)),
  SE = c(round(se_y_rd, 4), round(se_y_ld, 4)),
  CI_Lower = c(round(y_bar_rd - z_critical * se_y_rd, 4),
               round(ci_lower_ld, 4)),
  CI_Upper = c(round(y_bar_rd + z_critical * se_y_rd, 4),
               round(ci_upper_ld, 4))
)

print(comparison_table)
```

```
          Estimator Estimate  Variance      SE CI_Lower CI_Upper
              Ratio 400.8643 350.12679 18.7117 364.1900 437.5385
x_phase2 Regression 381.2944  26.14394  5.1131 371.2729 391.3159
```

```
cat("\nRelative Efficiency (Ratio vs Regression):",
    round(var_y_rd / var_y_ld_method1, 4), "\n")
```

```
Relative Efficiency (Ratio vs Regression): 13.3923
```

```
# Summary
cat("\n")
```

```
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```
cat("SUMMARY\n")
```

```
SUMMARY
```

```
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```
cat("Based on the two-phase regression estimation:\n")
```

```
Based on the two-phase regression estimation:
```

```
cat("  - Estimated average sleeping hours:", round(y_bar_ld, 2), "hours\n")
```

```
  - Estimated average sleeping hours: 381.29 hours
```

```
cat("  - We are 95% confident the true mean is between\n")
```

```
  - We are 95% confident the true mean is between
```

```
cat("    ", round(ci_lower_ld, 2), "and", round(ci_upper_ld, 2), "hours\n\n")
```

```
    371.27 and 391.32 hours
```

```r
cat("The regression estimator is generally more efficient than the ratio\n")
```

```
The regression estimator is generally more efficient than the ratio
```

```r
cat("estimator because it does not assume the relationship passes through\n")
```

```
estimator because it does not assume the relationship passes through
```

```r
cat("the origin (it includes an intercept term).\n")
```

```
the origin (it includes an intercept term).
```

Explanation:

Two-Phase Regression Estimator:

$$\bar{y}_{ld} = \bar{y} + \hat{\beta}(\bar{x}' - \bar{x})$$

where: - $\bar{y}$ = mean of y from second-phase sample (size n) - $\bar{x}$ = mean of x from second-phase sample (size n) - $\bar{x}'$ = mean of x from first-phase sample (size n') - $\hat{\beta} = \frac{s_{xy}}{s_x^2}$ = regression coefficient from phase 2

Variance Formula:

Method 1 (using correlation):

$$V(\bar{y}_{ld}) \approx \left(\frac{1}{n} - \frac{1}{n'}\right) s_y^2(1 - r_{xy}^2) + \frac{s_{x'}^2}{n'}$$

Method 2 (using MSE):

$$V(\bar{y}_{ld}) \approx \left(\frac{1}{n} - \frac{1}{n'}\right) MSE + \frac{s_{x'}^2}{n'}$$

where $MSE = \frac{1}{n-2}\sum(y_i - \hat{y}_i)^2$

Key Differences: Ratio vs Regression

Feature	Ratio Estimator	Regression Estimator
Model	$y = Rx$ (through origin)	$y = \alpha + \beta x$ (with intercept)
Efficiency	Good if relationship is proportional	Generally more efficient
Bias	Slightly biased	Approximately unbiased
Variance	Depends on R^2	Reduced by factor $(1 - r^2)$

Advantages of Regression:

- Accounts for intercept (more flexible)
- Generally lower variance
- Better when relationship doesn't pass through origin
- Variance reduction proportional to r^2 (squared correlation)

Q.(11) Select a first-phase sample of 20 units by SRSWOR from excel sheet Income and expenditure and note only the income of tribals in Western Ghats of Maharashtra. From the selected first-phase sample, select a subsample of 10 units and note income and expenditure. Estimate the average expenditure using ratio and regression estimators. Deduce their 95% confidence intervals.

```
# Load income and expenditure data
income_exp_data <- readxl::read_xlsx("data/Income and expenditure.xlsx")

# Display data structure
cat("Data structure:\n")
```

```
Data structure:
```

```
print(head(income_exp_data))
```

```
# A tibble: 6 x 3
  `Sr. No.` Income Expenditure
      <dbl>  <dbl>       <dbl>
1         1  24000        4700
2         2  25000        4000
3         3  31000        8500
4         4  21000        5000
5         5  27000        7000
6         6  35000        5500
```

```
cat("\n")
```

```
# Variables
x <- income_exp_data$Income        # Auxiliary variable (cheap to measure)
y <- income_exp_data$Expenditure   # Variable of interest (expensive to measure)

N <- length(y)  # Population size
n1 <- 20        # First-phase sample size
n2 <- 10        # Second-phase sample size

cat("TWO-PHASE SAMPLING: INCOME AND EXPENDITURE\n")
```

```
TWO-PHASE SAMPLING: INCOME AND EXPENDITURE
```

```r
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```r
cat("Population size (N):", N, "\n")
```

```
Population size (N): 40
```

```r
cat("First-phase sample size (n'):", n1, "\n")
```

```
First-phase sample size (n'): 20
```

```r
cat("Second-phase sample size (n):", n2, "\n\n")
```

```
Second-phase sample size (n): 10
```

```r
# PHASE 1: Select 20 units by SRSWOR and measure only Income
set.seed(456)
phase1_indices <- sample(1:N, n1, replace = FALSE)
x_phase1 <- x[phase1_indices]

cat("PHASE 1: Measure Income only\n")
```

```
PHASE 1: Measure Income only
```

```r
cat(strrep("-", 70), "\n")
```

```
----------------------------------------------------------------------
```

```r
cat("Selected indices:", phase1_indices, "\n\n")
```

```
Selected indices: 37 35 38 21 27 25 14 31 26 9 15 32 19 11 23 8 34 13 5 20
```

```r
x_bar_prime <- mean(x_phase1)
s_x_prime_sq <- var(x_phase1)

cat("First-phase mean of Income (x̄'):", round(x_bar_prime, 2), " \n\n")
```

```
First-phase mean of Income (x̄'): 49840
```

```r
# PHASE 2: Select 10 units from phase 1 and measure both
set.seed(457)
phase2_indices_within <- sample(1:n1, n2, replace = FALSE)
phase2_indices <- phase1_indices[phase2_indices_within]

x_phase2 <- x[phase2_indices]
y_phase2 <- y[phase2_indices]

cat("PHASE 2: Measure both Income and Expenditure\n")
```

```
PHASE 2: Measure both Income and Expenditure
```

```r
cat(strrep("-", 70), "\n")
```

```
----------------------------------------------------------------------
```

```r
cat("Selected indices from phase 1:", phase2_indices_within, "\n\n")
```

```
Selected indices from phase 1: 8 13 12 3 16 10 4 14 19 15
```

```r
x_bar <- mean(x_phase2)
y_bar <- mean(y_phase2)
s_y_sq <- var(y_phase2)
s_x_sq <- var(x_phase2)
s_xy <- cov(x_phase2, y_phase2)
r_xy <- cor(x_phase2, y_phase2)

cat("Second-phase mean of Income (x̄):", round(x_bar, 2), " \n")
```

```
Second-phase mean of Income (x̄): 66640
```

```r
cat("Second-phase mean of Expenditure (ȳ):", round(y_bar, 2), " \n")
```

```
Second-phase mean of Expenditure (ȳ): 21160
```

```r
cat("Correlation (r_xy):", round(r_xy, 4), "\n\n")
```

```
Correlation (r_xy): 0.1616
```

```r
# RATIO ESTIMATOR
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```r
cat("RATIO ESTIMATOR\n")
```

```
RATIO ESTIMATOR
```

```r
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```r
R_hat <- y_bar / x_bar
y_bar_rd <- R_hat * x_bar_prime

cat("Sample ratio R:", round(R_hat, 4), "\n")
```

```
Sample ratio R: 0.3175
```

```r
cat("Ratio estimate (ȳ_rd):", round(y_bar_rd, 2), " \n\n")
```

```
Ratio estimate (ȳ_rd): 15825.55
```

```r
term1_ratio <- (1/n2 - 1/n1) * (s_y_sq + R_hat^2 * s_x_sq - 2 * R_hat * s_xy)
term2_ratio <- (R_hat^2 * s_x_prime_sq) / n1
var_y_rd <- term1_ratio + term2_ratio
se_y_rd <- sqrt(var_y_rd)

cat("Variance V(ȳ_rd):", round(var_y_rd, 4), "\n")
```

```
Variance V(ȳ_rd): 111890518
```

```r
cat("Standard error:", round(se_y_rd, 2), " \n\n")
```

```
Standard error: 10577.83
```

```r
z_crit <- qnorm(0.975)
ci_lower_rd <- y_bar_rd - z_crit * se_y_rd
ci_upper_rd <- y_bar_rd + z_crit * se_y_rd

cat("95% CI (Ratio): [", round(ci_lower_rd, 2), ",",
    round(ci_upper_rd, 2), "]  \n\n")
```

```
95% CI (Ratio): [ -4906.62 , 36557.71 ]
```

```r
# REGRESSION ESTIMATOR
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```r
cat("REGRESSION ESTIMATOR\n")
```

```
REGRESSION ESTIMATOR
```

```r
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```r
model <- lm(y_phase2 ~ x_phase2)
beta_hat <- coef(model)[2]
alpha_hat <- coef(model)[1]

y_bar_ld <- y_bar + beta_hat * (x_bar_prime - x_bar)

cat("Regression slope (^):", round(beta_hat, 4), "\n")
```

```
Regression slope (^): 0.0389
```

```r
cat("Regression estimate (ȳ_ld):", round(y_bar_ld, 2), " \n\n")
```

```
Regression estimate (ȳ_ld): 20505.79
```

```r
var_y_ld <- (1/n2 - 1/n1) * s_y_sq * (1 - r_xy^2) + s_x_prime_sq / n1
se_y_ld <- sqrt(var_y_ld)

cat("Variance V(ȳ_ld):", round(var_y_ld, 4), "\n")
```

```
Variance V(ȳ_ld): 350900531
```

```r
cat("Standard error:", round(se_y_ld, 2), " \n\n")
```

```
Standard error: 18732.34
```

```r
ci_lower_ld <- y_bar_ld - z_crit * se_y_ld
ci_upper_ld <- y_bar_ld + z_crit * se_y_ld

cat("95% CI (Regression): [", round(ci_lower_ld, 2), ",",
    round(ci_upper_ld, 2), "]  \n\n")
```

```
95% CI (Regression): [ -16208.92 , 57220.5 ]
```

```
# COMPARISON
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```
cat("COMPARISON OF ESTIMATORS\n")
```

```
COMPARISON OF ESTIMATORS
```

```
cat(strrep("=", 70), "\n")
```

```
======================================================================
```

```
comparison_table <- data.frame(
  Estimator = c("Ratio", "Regression"),
  Estimate = c(round(y_bar_rd, 2), round(y_bar_ld, 2)),
  Variance = c(round(var_y_rd, 4), round(var_y_ld, 4)),
  SE = c(round(se_y_rd, 2), round(se_y_ld, 2)),
  CI_Lower = c(round(ci_lower_rd, 2), round(ci_lower_ld, 2)),
  CI_Upper = c(round(ci_upper_rd, 2), round(ci_upper_ld, 2))
)

print(comparison_table)
```

```
          Estimator Estimate  Variance       SE  CI_Lower CI_Upper
              Ratio 15825.55 111890518 10577.83  -4906.62 36557.71
x_phase2 Regression 20505.79 350900531 18732.34 -16208.92 57220.50
```

```
cat("\nRelative Efficiency:", round(var_y_rd / var_y_ld, 4), "\n")
```

```
Relative Efficiency: 0.3189
```

```
cat("The", ifelse(var_y_ld < var_y_rd, "regression", "ratio"),
    "estimator is more efficient.\n")
```

```
The ratio estimator is more efficient.
```

Explanation:

Two-Phase Design for Income and Expenditure:

- **Phase 1 (n' = 20):** Measure Income only (cheap survey)
- **Phase 2 (n = 10):** Measure both Income and Expenditure (expensive detailed survey)

Estimators:

Ratio: $\bar{y}_{rd} = \hat{R} \times \bar{x}' = \frac{\bar{y}}{\bar{x}} \times \bar{x}'$

Regression: $\bar{y}_{ld} = \bar{y} + \hat{\beta}(\bar{x}' - \bar{x})$

Practical Application:

This design is common in socio-economic surveys where: - Income data is easier to collect (records, simplified questions) - Expenditure data requires detailed tracking and recall - Strong correlation between income and expenditure improves efficiency