Problem 01: We are interested to know the relative change over a two-year period

in the assessed value of homes in a certain community. We are conducting a simple survey sampling of n = 20 homes from the N = 1000 total homes in the community. We obtain the values for this year(y) and the corresponding values from two years ago(x) for each of the n = 20 homes include in the sample that given below.

x	6.7, 8.2, 7.9, 6.4, 8.3, 7.2, 6.0, 7.4, 8.1, 9.3, 8.2, 6.8, 7.4, 7.5, 8.3, 9.1, 8.6, 7.9, 6.3, 8.9
y	7.1, 8.4, 8.2, 6.9, 8.4, 7.9, 6.5, 7.6, 8.9, 9.9, 9.1, 7.3, 7.8, 8.9, 9.6, 8.7, 8.8, 7.0, 9.4

Estimate the relative change(R) and the ratio estimator of population mean in the assessed values for the N = 1000 homes by using X = 7.8 ; also find the estimated variance of these estimators.

Solution:

```
x <- c(6.7, 8.2, 7.9, 6.4, 8.3, 7.2, 6.0, 7.4, 8.1, 9.3, 8.2, 6.8, 7.4, 7.5, 8.3, 9.1, 8.6, 7.9, 6.3, 8.9)
y <- c(7.1, 8.4, 8.2, 6.9, 8.4, 7.9, 6.5, 7.6, 8.9, 9.1, 7.3, 8.3, 8.9, 9.6, 8.7, 8.8, 8.0, 7.0, 9.4)
N <- 1000
Xbar <- 7.8
ratio <- function(x, y, N) {
n <- length(x)
r <- sum(y) / sum(x)
var.r <- (N - n) / (n * (n - 1) * N * Xbar^2) * sum((y - r * x)^2)
ztab <- qnorm(0.05 / 2, lower.tail = FALSE)
LCL <- r - ztab * sqrt(var.r)
UCL <- r + ztab * sqrt(var.r)
return(cbind(ratio = r, variance.ratio = var.r, LCL = LCL, UCL = UCL))
```

Output:

```
     ratio    variance.ratio    LCL       UCL
[1,] 1.003236  0.001089932   0.9385298 1.067943
```

Comments: The relative change (R ≈ 0.06601) suggests that, on average, current home values are 97.5% of their levels two years ago, indicating a slight decline in assessed values. The estimated population mean (≈ 7.84) represents the average current assessed value per home. Once calculated, the variance will aid in constructing confidence intervals, providing insight into the reliability of the estimate.

Problem 02:

A list of 23 farmer agricultural districts of Bangladesh with areas in thousand areas of lands (x) is given in the accompanying table together with district wise production of rice (y) (in 1000 metric tons) in 1998 – 99.

District	x	y	District	x	y
1.Banderban	61	48	13.Tangail	561	483
2.Chattogram	1079	994	14.Barishal	133	662
3.Khagrachari	30	26	15.Jessore	1452	1352
4.Cumilla	1519	1313	16.Khulna	1134	853
5.Noakhali	1036	779	17.Kustia	567	479
6.Rangamati	48	40	18.Patuakhali	1027	543
7.Sylhet	2309	1512	19.Bogra	1169	1093
8.Dhaka	936	859	20.Dinajpur	1573	1069
9.Faidpur	1018	577	21.Pabna	738	660
10.Jamalpur	811	723	22.Rajshahi	1799	1753
11.Kishorgonj	1341	1121	23.Rangpur	2243	1873
12.Mymensingh	1715	928	Total	24299	19740

a) Draw a sample of 5 districts without replacement using,

(i) Simple random sampling.

(ii) PPS method.

b) Estimate the average and total production of rice per district for both the samples.

c) Compute the 95% confidence interval in each case.

Solution:

```
X <- c(61, 1079, 30, 1519, 1036, 48, 2309, 936, 1018, 811, 1341, 1715, 561, 133, 1452, 1134, 567, 1027, 1169, 1573, 738, 1799, 2243)
Y <- c(48, 994, 26, 1313, 779, 40, 1512, 859, 577, 723, 1121, 928, 483, 662, 1352, 853, 479, 543, 1093, 1069, 660, 1753, 1873)
N <- length(Y)
n <- 5
# a) SRSWOR Function
SRSWOR <- function(X, Y, n) {
y <- c(483,723,40,1093,660)
ybar <- mean(y)
Ybar <- mean(Y)
S_sq <- sum((Y - Ybar)^2) / (N - 1)
s_sq <- sum((y - ybar)^2) / (n - 1)
var.ybar <- (N - n)/N * (s_sq / n)
se.ybar  <- sqrt(var.ybar)
# 95% confidence interval
ztab <- qnorm(0.05/2, lower.tail = FALSE)
LCL <- ybar - ztab * se.ybar
UCL <- ybar + ztab * se.ybar
Y.hat <- N * ybar
return(rbind(ybar = ybar, Ybar = Ybar, Y.hat = Y.hat, S_sq = S_sq, s_sq = s_sq, var.ybar = var.ybar, se.ybar = se.ybar, LCL = LCL, UCL = UCL))
}


SRSWOR(X, Y, n)
```

```r
# b) PPS Sampling (Hansen–Hurwitz)
cbind(X = X, Cum.sum = cumsum(X), Probability = round(X / sum(X), 3))
ran <- c(15,12,20,8,22)     # Replace with your PPS sample indices
yi <- Y[ran]
xi <- X[ran]
cbind(yi, xi)
Xt <- sum(X)
# PPS estimates
ybar_pps <- Xt / (N * n) * sum(yi / xi)
Yhat_pps <- N * ybar_pps
var.ybar_pps <- sum((Xt/N * (yi/xi) - ybar_pps)^2) / (n * (n - 1))
Se.ybar_pps  <- sqrt(var.ybar_pps)
# 95% confidence interval
ztab <- qnorm(0.05/2, lower.tail = FALSE)
LCL <- ybar_pps - ztab * Se.ybar_pps
UCL <- ybar_pps + ztab * Se.ybar_pps
# Output PPS results
cbind(ybar_pps = ybar_pps, Yhat_pps = Yhat_pps, var.ybar_pps = var.ybar_pps, LCL = LCL, UCL = UCL)
```

Output:

ybar_pps	Yhat_pps	var.ybar_pps	LCL	UCL
854.4787	19653.01	7959.139	679.6225	1029.335

ybar	599.8000
Ybar	858.2609
Y.hat	13795.4000
S_sq	249288.5652
s_sq	147266.7000
var.ybar	23050.4400
se.ybar	151.8237
LCL	302.2310
UCL	897.3690

Problem 03: You are tasked with estimating the average daily calorie intake of individuals in a population across different age groups. To achieve this, you decided to employ double sampling, a cost-effective sampling technique.

a) Implement the double sampling technique with the following specifications:

Select 50 individuals for each sampling day.

Sample data for 5 days.

b) Write a function to perform double sampling on the dataset, randomly selecting a subset of individuals for each day and a subset of days to repeat the process.

c) Calculate the average daily calorie intake from the sampled data and estimate the overall average for the entire population.

d) Discuss the implications of using double sampling in estimating population parameters and provide insights based on your findings.

Solution:

a)

```
data = read.csv(file.choose())
data
```

b)

```
double_sampling = function(data, i, d) {
set.seed(123)
individual = sample(data$ID, i, replace=FALSE)
day = sample(data$ID, d, replace=FALSE)
sampled_data = data[individual[day], ]
return(sampled_data)
}
sampled_data = double_sampling(data=data, i=50, d=5)
sampled_data
```

Output:

	ID	Age	Calories
40	40	48	2300
41	41	40	2350
24	24	55	3000
32	32	47	2700
17	7	40	2500

c)

```
ybar = mean(sampled_data$Calories)
ybar
Ybar = mean(data$Calories)
Ybar
```

Output:

```
2570

2550
```

d)

```
cat("The average daily calorie intake from double sampling data is =", ybar, "\n")
cat("The overall average daily calorie intake for entire population is =", Ybar, "\n")
cat("The moderate distance between them is =", abs(ybar - Ybar), "\n")
```

Output:

```
The average daily calorie intake from double sampling data is = 2570
The overall average daily calorie intake for entire population is = 2550
The moderate distance between them is = 20
```

Comment: The distance is small, Here the double sampling estimator is a good representative of population parameter.

Problem 04:

Suppose you have 4 schools, each with 3 classes, and each class has 10 students.

- Select 2 schools (first stage), then 1 class per selected school (second stage), then 5 students per selected class (third stage).
- Estimate the **average score** of students if each student has a math score generated randomly from N (50, 10^2).

Solution:

```
set.seed(123)
schools <- 1:4
classes <- 1:3
students <- 1:10
population <- expand.grid(School = schools, Class = classes, Student = students)
population$Score <- round(rnorm(nrow(population), mean = 50, sd = 10), 1)
# Stage 1: Select 2 schools
selected_schools <- sample(schools, 2)
# Stage 2: Select 1 class per selected school
selected_classes <- unlist(lapply(selected_schools, function(s) sample(classes, 1)))
# Stage 3: Select 5 students per selected class (fixed)
sampled_data <- do.call(rbind, lapply(1:2, function(i) {
df <- subset(population, School == selected_schools[i] & Class == selected_classes[i])
df[sample(nrow(df), 5), ]
}))
sampled_data
```

Comment:

This is a classic **multi-stage sampling** design:

- First stage: clusters of schools.
- Second stage: clusters of classes within selected schools.
- Third stage: individual units (students) within selected classes.

Output:

	School	Class	Student	Score
1	1	1	1	44.4
85	1	1	8	47.8
97	1	1	9	71.9
13	1	1	2	54.0
109	1	1	10	46.2
94	2	3	8	43.7
22	2	3	2	47.8
58	2	3	5	55.8
118	2	3	10	43.6
10	2	3	1	45.5

Problem 05:

Assume we want to estimate the **average household income**.

- First, take a simple random sample of 100 households to collect **auxiliary information** (e.g., household size).
- Then, take a subsample of 40 households to collect **income data**.
- Use **regression estimation** to improve the income estimate.

Solution:

```
set.seed(123)
N <- 1000
household_size <- rpois(N, 5) + 1
income <- 2000 * household_size + rnorm(N, mean = 0, sd = 5000)
phase1 <- sample(1:N, 100)
aux_data <- data.frame(ID = phase1, Size = household_size[phase1])
phase2 <- sample(phase1, 40)
sub_data <- data.frame(ID = phase2, Size = household_size[phase2], Income = income[phase2])
reg_model <- lm(Income ~ Size, data = sub_data)
y_hat <- predict(reg_model, newdata = data.frame(Size = household_size))
reg_estimate <- mean(y_hat)
true_mean <- mean(income)
c(True = true_mean, Regression_Estimate = reg_estimate)
```

Output:

True	12021.6650875515
Regression_Estimate	12751.9945445453

Problem 06:

Simulate a population of 1000 people with a binary variable Smoker (1 = yes, 0 = no, probability = 0.3).

- Draw 50 SRS samples of size 100.
- For each sample, compute the proportion of smokers.
- Plot the distribution of sample proportions and compare it with the true population proportion.
- Calculate **bias and mean squared error (MSE)**.

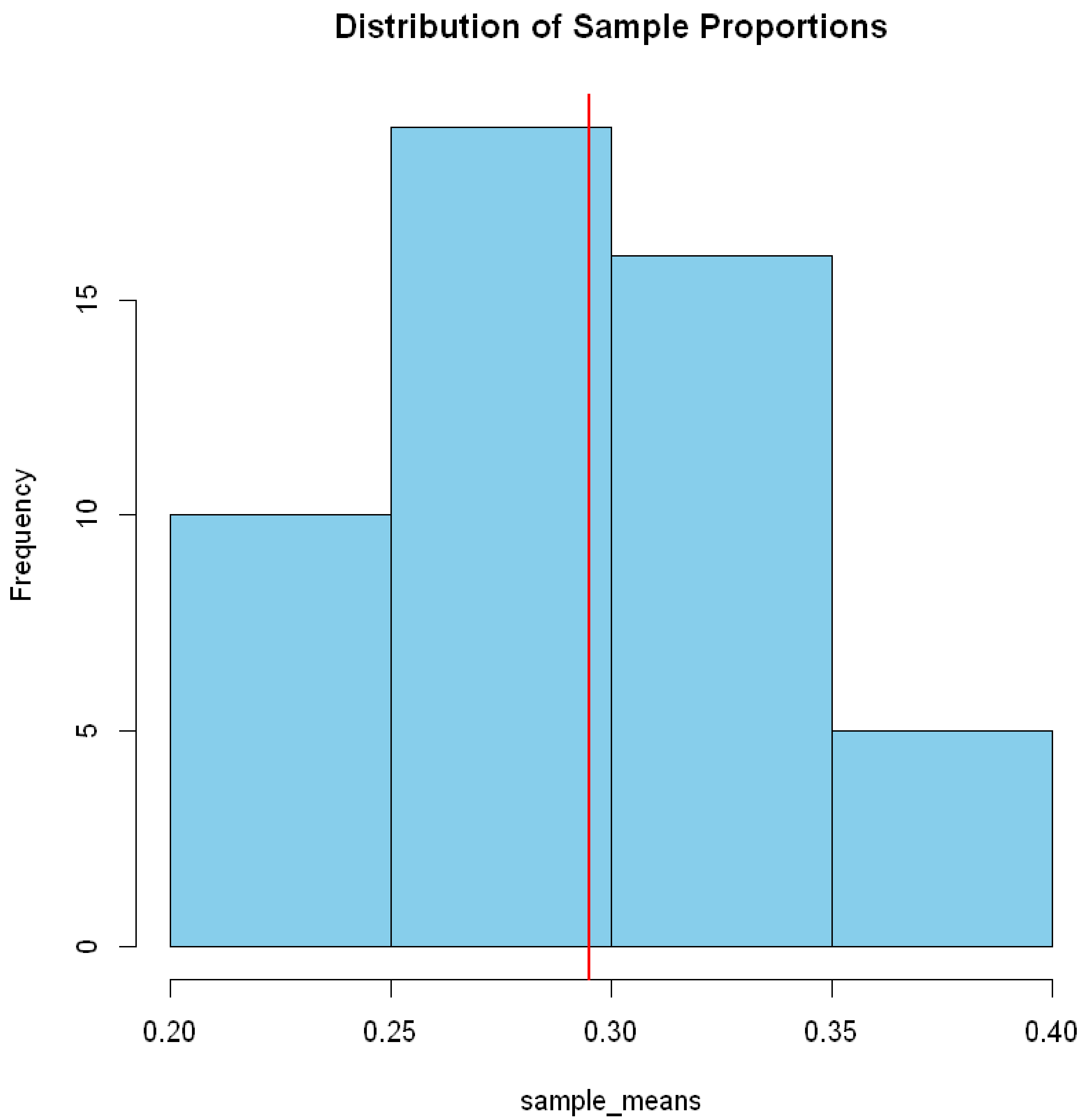
Solution:

```
set.seed(123)
N <- 1000
Smoker <- rbinom(N, 1, 0.3)
simulate_srs <- function() {
samp <- sample(Smoker, 100)
mean(samp)
}
sample_means <- replicate(50, simulate_srs())
bias <- mean(sample_means) - mean(Smoker)
mse <- mean((sample_means - mean(Smoker))^2)
hist(sample_means, col = "skyblue", main = "Distribution of Sample Proportions")
abline(v = mean(Smoker), col = "red", lwd = 2)
c(True_Proportion = mean(Smoker), Bias = bias, MSE = mse)
```

Output:

True_Proportion	0.295
Bias	0.00180000000000002
MSE	0.001969

Histogram:



Comment: This code simulates taking many SRS samples from a population with 30% smokers, computes the sample proportion each time, and then evaluates the estimator's bias and mean squared error to check how good the SRS estimate is.

Problem 07:

Generate a population of 500 households with an expenditure variable (rnorm(500, mean=15000, sd=3000)).

- Compute the true mean expenditure (census).
- Take a SRS sample of size 50 and compute the sample mean.
- Repeat 1000 times and plot the distribution of sample means.
- Comment on how well surveys approximate census results.

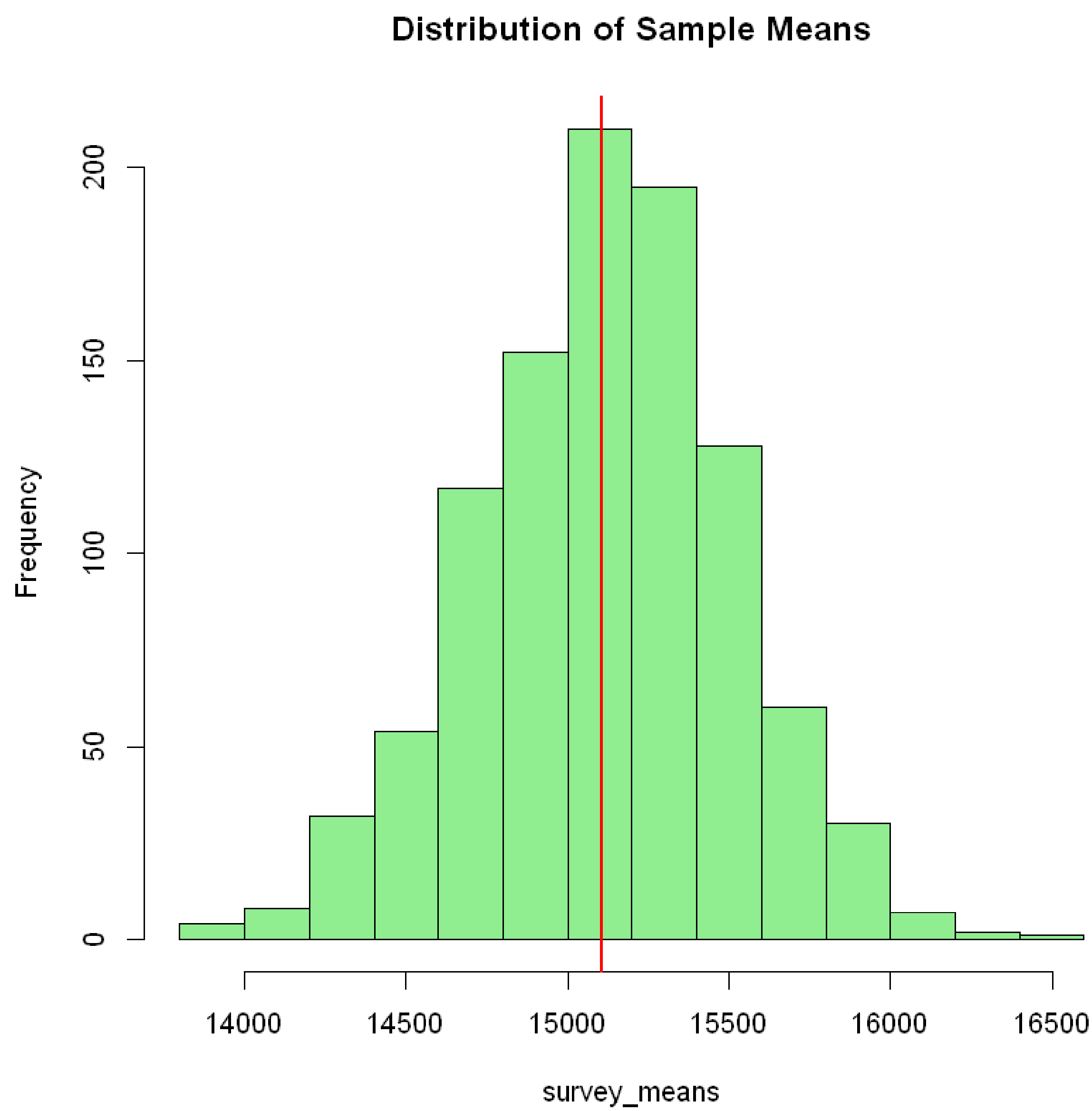
Solution:

```
set.seed(123)
population <- rnorm(500, mean = 15000, sd = 3000)
true_mean <- mean(population)
survey_mean <- function() {
samp <- sample(population, 50)
mean(samp)
}
survey_means <- replicate(1000, survey_mean())
hist(survey_means, col = "lightgreen", main = "Distribution of Sample Means")
abline(v = true_mean, col = "red", lwd = 2)
c(True_Census_Mean = true_mean, Survey_Mean = mean(survey_means))
```

Output:

True_Census_Mean	15103.7713425869
Survey_Mean	15110.9509187104

Histogram:



Comment: This code simulates repeatedly sampling 50 households from a population of 500, computes the sample mean each time, and shows how the distribution of those sample means clusters around the true census mean — illustrating sampling variability and unbiasedness of the sample mean.