480 project

2025-04-18

R Markdown

```
set.seed(123)

# clean data
data_clean <- read_sav("Downloads/ATP W131.sav") %>%
  select(EVERLEAD1_W131, F_GENDER, F_AGECAT, WEIGHT_W131) %>%
  mutate(
    across(everything(), ~ ifelse(. == 99, NA, .)),
    leader = ifelse(EVERLEAD1_W131 <= 3, 1, 0),
    F_GENDER = factor(F_GENDER, levels = c(1, 2), labels = c("Male", "Female")),
    F_AGECAT = factor(F_AGECAT, levels = c(1, 2, 3, 4),
                      labels = c("18-29", "30-49", "50-64", "65+"))
  ) %>%
  drop_na() %>%
  filter(F_GENDER %in% c("Male", "Female"))

# survey design
log_design <- svydesign(
  ids = ~1,
  weights = ~WEIGHT_W131,
  data = data_clean
)

# logistic regression model
log_reg <- svyglm(
  leader ~ F_GENDER + F_AGECAT,
  design = log_design,
  family = quasibinomial()
)

summary(log_reg)
```

```
##
## Call:
## svyglm(formula = leader ~ F_GENDER + F_AGECAT, design = log_design,
##     family = quasibinomial())
##
## Survey design:
## svydesign(ids = ~1, weights = ~WEIGHT_W131, data = data_clean)
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)     0.55789    0.11185   4.988 6.31e-07 ***
## F_GENDERFemale -0.24379    0.07172  -3.399 0.000681 ***
## F_AGECAT30-49  -0.10055    0.11916  -0.844 0.398831
## F_AGECAT50-64  -0.36723    0.12155  -3.021 0.002531 **
## F_AGECAT65+    -0.39015    0.12321  -3.166 0.001553 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 1.000245)
##
## Number of Fisher Scoring iterations: 4
```

```
library(dplyr)
library(haven)
library(survey)
library(tidyr)

# Load and clean data
data <- read_sav("Downloads/ATP W131.sav")

data_clean <- data |>
  as_tibble() |>
  dplyr::select(EVERLEAD1_W131, F_GENDER, F_AGECAT, WEIGHT_W131) |>
  mutate(across(everything(), ~ifelse(. == 99, NA, .))) |>
  drop_na() |>
  mutate(
    leader = ifelse(EVERLEAD1_W131 <= 3, 1, 0),
    F_GENDER = factor(F_GENDER),
    F_AGECAT = factor(F_AGECAT),
    stratum = interaction(F_GENDER, F_AGECAT, drop = TRUE)
  )

# Not enough observations per stratum if we include gender = 3
table(data_clean$stratum)
```

```
##
## 1.1 2.1 3.1 1.2 2.2 3.2 1.3 2.3 3.3 1.4 2.4 3.4
## 304 306  24 979 756  19 833 569   4 681 545   2
```

Revised version excluding gender = 3

```
library(haven)
library(dplyr)
library(survey)
library(tidyr)
library(purrr)
library(ggplot2)

# Load the data
data <- read_sav("Downloads/ATP W131.sav")
```

```r
# Clean the data and select relevant variables
data_clean <- data |>
  as_tibble() |>
  dplyr::select(EVERLEAD1_W131, F_GENDER, F_AGECAT, WEIGHT_W131) |>
  mutate(across(everything(), ~ifelse(. == 99, NA, .))) |> # Convert 99 to NA
  drop_na() |>
# Only including male and female for gender variable
  filter(F_GENDER %in% c(1, 2)) |>
  mutate(
# Convert leadership experience variable into binary(1-3: Yes, 4-5: No)
    leader = ifelse(EVERLEAD1_W131 <= 3, 1, 0),
    F_GENDER = factor(F_GENDER, labels = c("Male", "Female")),
    F_AGECAT = factor(F_AGECAT, labels = c("18-29", "30-49", "50-64", "65+")),
# Define stratum
    stratum = interaction(F_GENDER, F_AGECAT, drop = TRUE)
  )


# Check if we have enough number of observations per stratum
table(data_clean$stratum)
```

```
##
##   Male.18-29 Female.18-29   Male.30-49 Female.30-49   Male.50-64 Female.50-64
##          304          306          979          756          833          569
##     Male.65+   Female.65+
##          681          545
```

```r
# Stratified Random Sampling Simulation of size 400
set.seed(123)
stratum_info <- data_clean |>
  count(stratum) |>
  mutate(prop = n / sum(n),               # Proportion of each stratum
         sample_n = round(prop * 400))    # Sample size per stratum

# Merge the sample size info to original data
data_joined <- data_clean |>
  inner_join(stratum_info, by = "stratum")
data_joined
```

```
## # A tibble: 4,973 x 9
##    EVERLEAD1_W131 F_GENDER F_AGECAT WEIGHT_W131 leader stratum          n   prop
##             <dbl> <fct>    <fct>          <dbl>  <dbl> <fct>        <int>  <dbl>
##  1              4 Male     65+            1.14       0 Male.65+       681 0.137
##  2              2 Male     50-64          0.280      1 Male.50-64     833 0.168
##  3              1 Female   18-29          1.23       1 Female.18-29   306 0.0615
##  4              3 Male     65+            0.960      1 Male.65+       681 0.137
##  5              2 Male     65+            0.413      1 Male.65+       681 0.137
##  6              3 Male     50-64          1.13       1 Male.50-64     833 0.168
##  7              4 Female   65+            0.916      0 Female.65+     545 0.110
##  8              5 Male     65+            0.940      0 Male.65+       681 0.137
##  9              3 Female   50-64          1.25       1 Female.50-64   569 0.114
## 10              4 Male     50-64          1.14       0 Male.50-64     833 0.168
## # i 4,963 more rows
```

```
## # i 1 more variable: sample_n <dbl>
```

```
# Randomly sample within each stratum
strat_sample <- data_joined |>
  group_split(stratum) |>
  map_df(~ slice_sample(.x, n = .x$sample_n[1]))

# Define stratified design
design_strat <- svydesign(
  ids = ~1,
  strata = ~stratum,
  data = strat_sample,
  weights = ~WEIGHT_W131
)

# Estimate overall population mean using Stratified random sampling
svymean(~leader, design_strat)
```

```
##           mean     SE
## leader 0.58447 0.0304
```

```
svyby(~leader, ~stratum, design_strat, svymean)
```

```
##                   stratum    leader         se
## Male.18-29     Male.18-29 0.7540087 0.10256180
## Female.18-29 Female.18-29 0.7507892 0.09517137
## Male.30-49     Male.30-49 0.5980667 0.07827110
## Female.30-49 Female.30-49 0.5341338 0.06825218
## Male.50-64     Male.50-64 0.6247218 0.08850867
## Female.50-64 Female.50-64 0.4272505 0.07482393
## Male.65+         Male.65+ 0.6042135 0.08821080
## Female.65+     Female.65+ 0.4984795 0.08204961
```

```
# Logistic regression
strat_model <- svyglm(leader ~ F_GENDER + F_AGECAT,
                      design = design_strat,
                      family = quasibinomial())
summary(strat_model)
```

```
##
## Call:
## svyglm(formula = leader ~ F_GENDER + F_AGECAT, design = design_strat,
##     family = quasibinomial())
##
## Survey design:
## svydesign(ids = ~1, strata = ~stratum, data = strat_sample, weights = ~WEIGHT_W131)
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.3261     0.4044   3.279  0.00114 **
## F_GENDERFemale  -0.4071     0.2531  -1.609  0.10851
## F_AGECAT30-49   -0.8554     0.4286  -1.996  0.04666 *
```

```
## F_AGECAT50-64   -1.0029     0.4419  -2.270  0.02377 *
## F_AGECAT65+     -0.9156     0.4449  -2.058  0.04026 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 1.004276)
##
## Number of Fisher Scoring iterations: 4
```

```
# SRS
set.seed(123)
srs_sample <- data_clean |>
  slice_sample(n = nrow(strat_sample), replace = FALSE)

design_srs <- svydesign(ids = ~1, data = srs_sample, weights = ~WEIGHT_W131)

# Estimate proportion in SRS
svymean(~leader, design_srs)
```

```
##           mean     SE
## leader 0.60272 0.0294
```

```
# Compare standard error between Stratified sampling and SRS
se_compare <- tibble(
  method = c("Stratified Sampling", "SRS"),
  estimate = c(coef(svymean(~leader, design_strat)),
               coef(svymean(~leader, design_srs))),
  se = c(SE(svymean(~leader, design_strat)),
         SE(svymean(~leader, design_srs)))
)
se_compare
```

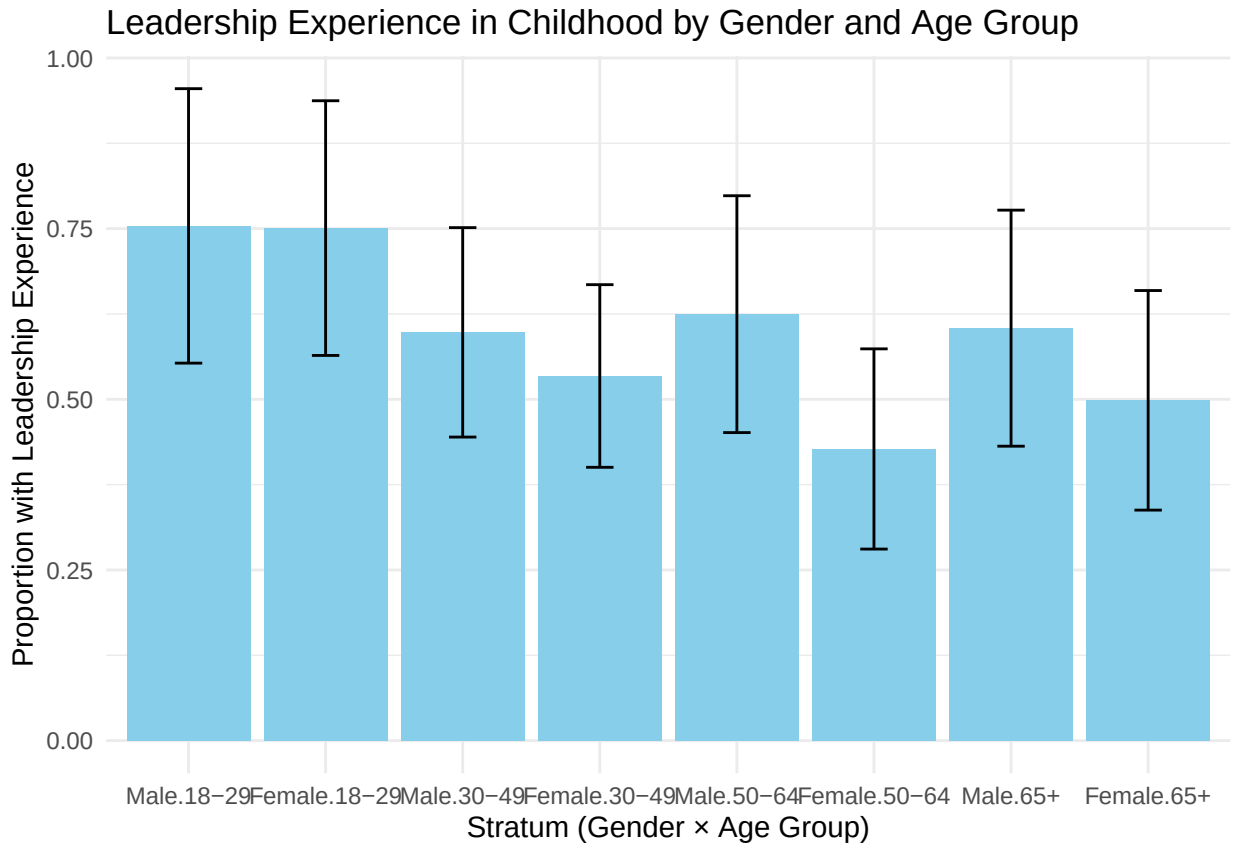
```
## # A tibble: 2 x 3
##   method              estimate     se
##   <chr>                  <dbl>  <dbl>
## 1 Stratified Sampling    0.584 0.0304
## 2 SRS                    0.603 0.0294
```

```
# Calculate 95% Confidence Intervals

leader_by_stratum <- svyby(~leader, ~stratum, design_strat, svymean)
leader_by_stratum <- leader_by_stratum |>
  mutate(
    lower = leader - 1.96 * se,
    upper = leader + 1.96 * se
  )
leader_by_stratum
```

```
##                   stratum    leader         se     lower     upper
## Male.18-29     Male.18-29 0.7540087 0.10256180 0.5529875 0.9550298
## Female.18-29 Female.18-29 0.7507892 0.09517137 0.5642533 0.9373251
## Male.30-49     Male.30-49 0.5980667 0.07827110 0.4446553 0.7514781
```

```
## Female.30-49 Female.30-49 0.5341338 0.06825218 0.4003595 0.6679081
## Male.50-64     Male.50-64 0.6247218 0.08850867 0.4512448 0.7981988
## Female.50-64 Female.50-64 0.4272505 0.07482393 0.2805956 0.5739054
## Male.65+         Male.65+ 0.6042135 0.08821080 0.4313203 0.7771066
## Female.65+     Female.65+ 0.4984795 0.08204961 0.3376623 0.6592968
```

```r
# Visualization

ggplot(leader_by_stratum, aes(x = stratum, y = leader)) +
  geom_col(fill = "skyblue") +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.2) +
  labs(
    title = "Leadership Experience in Childhood by Gender and Age Group",
    x = "Stratum (Gender × Age Group)",
    y = "Proportion with Leadership Experience"
  ) +
  theme_minimal()
```



Monte Carlo Version

```r
library(haven)
library(dplyr)
library(survey)
```

```r
library(tidyr)
library(purrr)
library(ggplot2)

# Load the data
data <- read_sav("Downloads/ATP W131.sav")

# Clean the data and select relevant variables
data_clean <- data |>
  as_tibble() |>
  dplyr::select(EVERLEAD1_W131, F_GENDER, F_AGECAT, WEIGHT_W131) |>
  mutate(across(everything(), ~ifelse(. == 99, NA, .))) |> # Convert 99 to NA
  drop_na() |>
# Only including male and female for gender variable
  filter(F_GENDER %in% c(1, 2)) |>
  mutate(
# Convert leadership experience variable into binary(1-3: Yes, 4-5: No)
    leader = ifelse(EVERLEAD1_W131 <= 3, 1, 0),
    F_GENDER = factor(F_GENDER, labels = c("Male", "Female")),
    F_AGECAT = factor(F_AGECAT, labels = c("18-29", "30-49", "50-64", "65+")),
# Define stratum
    stratum = interaction(F_GENDER, F_AGECAT, drop = TRUE)
  )


# Check if we have enough number of observations per stratum
table(data_clean$stratum)
```

```
##
##   Male.18-29 Female.18-29   Male.30-49 Female.30-49   Male.50-64 Female.50-64
##          304          306          979          756          833          569
##     Male.65+   Female.65+
##          681          545
```

```r
# Stratified Random Sampling Simulation of size 400
set.seed(123)
stratum_info <- data_clean |>
  count(stratum) |>
  mutate(prop = n / sum(n),               # Proportion of each stratum
         sample_n = round(prop * 400))    # Sample size per stratum


# Merge the sample size info to original data
data_joined <- data_clean |>
  inner_join(stratum_info, by = "stratum")
data_joined
```

```
## # A tibble: 4,973 x 9
##    EVERLEAD1_W131 F_GENDER F_AGECAT WEIGHT_W131 leader stratum          n   prop
##             <dbl> <fct>    <fct>          <dbl>  <dbl> <fct>        <int>  <dbl>
##  1              4 Male     65+            1.14       0 Male.65+       681 0.137
##  2              2 Male     50-64          0.280      1 Male.50-64     833 0.168
##  3              1 Female   18-29          1.23       1 Female.18-29   306 0.0615
```

```
##  4              3 Male     65+            0.960      1 Male.65+       681 0.137
##  5              2 Male     65+            0.413      1 Male.65+       681 0.137
##  6              3 Male     50-64          1.13       1 Male.50-64     833 0.168
##  7              4 Female   65+            0.916      0 Female.65+     545 0.110
##  8              5 Male     65+            0.940      0 Male.65+       681 0.137
##  9              3 Female   50-64          1.25       1 Female.50-64   569 0.114
## 10              4 Male     50-64          1.14       0 Male.50-64     833 0.168
## # i 4,963 more rows
## # i 1 more variable: sample_n <dbl>
```

```r
# Define stratified design
design_strat <- svydesign(
  ids = ~1,
  strata = ~stratum,
  data = strat_sample,
  weights = ~1
)

# Estimate overall population mean using Stratified random sampling
svymean(~leader, design_strat)
```

```
##           mean     SE
## leader 0.59352 0.0245
```

```r
svyby(~leader, ~stratum, design_strat, svymean)
```

```
##                   stratum    leader         se
## Male.18-29     Male.18-29 0.5833333 0.10279899
## Female.18-29 Female.18-29 0.7600000 0.08717798
## Male.30-49     Male.30-49 0.6329114 0.05457699
## Female.30-49 Female.30-49 0.5737705 0.06384329
## Male.50-64     Male.50-64 0.6417910 0.05901917
## Female.50-64 Female.50-64 0.4565217 0.07425327
## Male.65+         Male.65+ 0.6000000 0.06666667
## Female.65+     Female.65+ 0.5227273 0.07617047
```

```r
# Logistic regression
strat_model <- svyglm(leader ~ F_GENDER + F_AGECAT,
                      design = design_strat,
                      family = quasibinomial())
summary(strat_model)
```

```
##
## Call:
## svyglm(formula = leader ~ F_GENDER + F_AGECAT, design = design_strat,
##     family = quasibinomial())
##
## Survey design:
## svydesign(ids = ~1, strata = ~stratum, data = strat_sample, weights = ~1)
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)      0.8747     0.3331   2.626  0.00898 **
## F_GENDERFemale  -0.2884     0.2073  -1.391  0.16489
## F_AGECAT30-49   -0.3116     0.3546  -0.879  0.38019
## F_AGECAT50-64   -0.4890     0.3629  -1.347  0.17868
## F_AGECAT65+     -0.4811     0.3705  -1.298  0.19498
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 1.003589)
##
## Number of Fisher Scoring iterations: 4
```

```
# SRS
set.seed(123)
srs_sample <- data_clean |>
  slice_sample(n = nrow(strat_sample), replace = FALSE)

design_srs <- svydesign(ids = ~1, data = srs_sample, weights = ~1)

# Estimate proportion in SRS
svymean(~leader, design_srs)
```

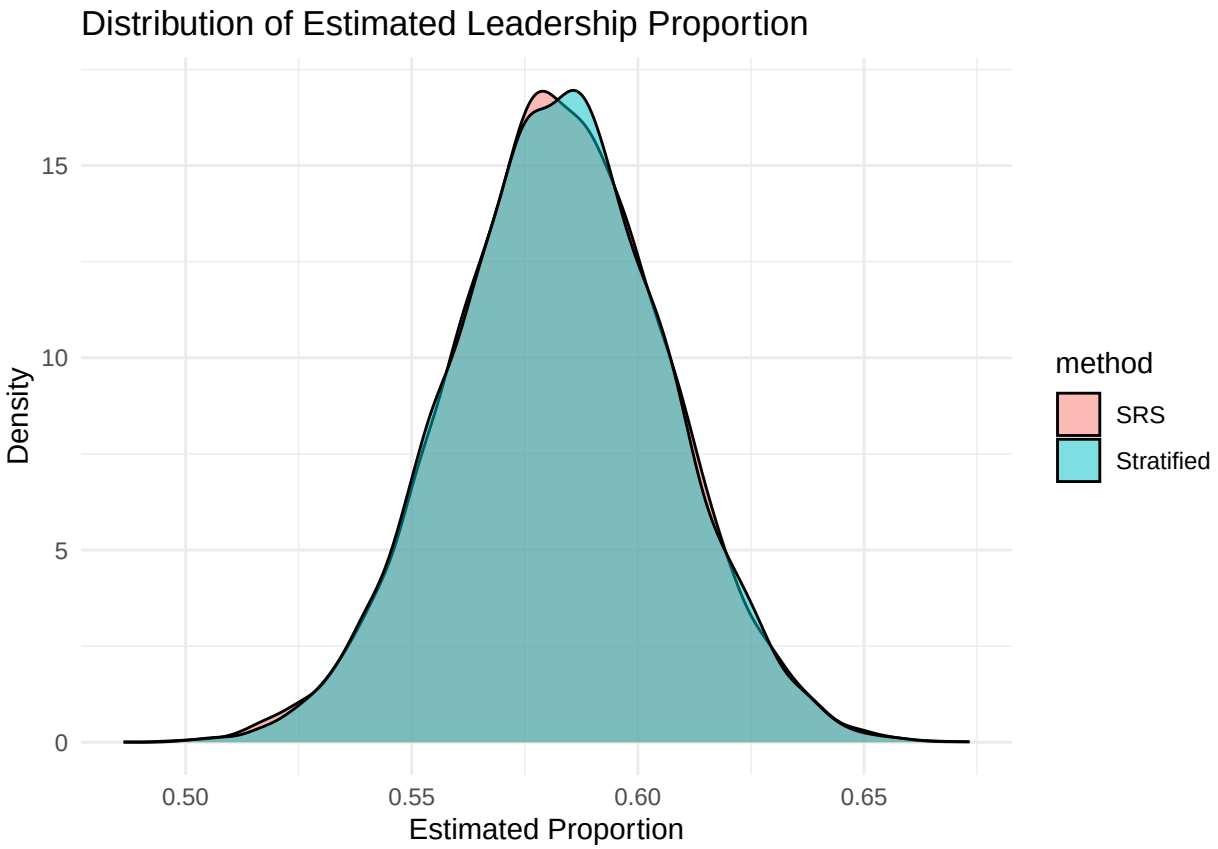
```
##          mean    SE
## leader 0.6384 0.024
```

```
# Compare standard error between Stratified sampling and SRS
se_compare <- tibble(
  method = c("Stratified Sampling", "SRS"),
  estimate = c(coef(svymean(~leader, design_strat)),
               coef(svymean(~leader, design_srs))),
  se = c(SE(svymean(~leader, design_strat)),
         SE(svymean(~leader, design_srs)))
)
se_compare
```

```
## # A tibble: 2 x 3
##   method              estimate     se
##   <chr>                  <dbl>  <dbl>
## 1 Stratified Sampling    0.594 0.0245
## 2 SRS                    0.638 0.0240
```

```
# Monte Carlo simulation of stratified sampling
n_iter <- 10000
strat_results <- replicate(n_iter, {
  strat_sample <- data_joined |>
    group_split(stratum) |>
    map_df(~ slice_sample(.x, n = .x$sample_n[1]))

  design <- svydesign(ids = ~1, strata = ~stratum, data = strat_sample, weights = ~1)
  est <- coef(svymean(~leader, design))
  se <- SE(svymean(~leader, design))
  c(estimate = est, se = se)
```

```r
})

# Monte Carlo simulation of SRS
set.seed(123)
n_srs <- nrow(data_joined |> group_split(stratum) |> map_df(~ slice_sample(.x, n = .x$sample_n[1])))

srs_results <- replicate(n_iter, {
  srs_sample <- data_clean |>
    slice_sample(n = n_srs, replace = FALSE)

  design <- svydesign(ids = ~1, data = srs_sample, weights = ~1)
  est <- coef(svymean(~leader, design))
  se <- SE(svymean(~leader, design))
  c(estimate = est, se = se)
})

# Create results data frame
strat_df <- as.data.frame(t(strat_results))
colnames(strat_df) <- c("estimate", "se")
strat_df$method <- "Stratified"

# After replicating SRS results
srs_df <- as.data.frame(t(srs_results))
colnames(srs_df) <- c("estimate", "se")
srs_df$method <- "SRS"

# Combine
results <- rbind(strat_df, srs_df)

# Visualization
ggplot(results, aes(x = estimate, fill = method)) +
  geom_density(alpha = 0.5) +
  labs(title = "Distribution of Estimated Leadership Proportion",
       x = "Estimated Proportion", y = "Density") +
  theme_minimal()
```



Stratified sampling method yielded a narrower and more concentrated distribution compared to SRS, demonstrating greater precision.