

plsc30600_section_1_solution.R

moffer

2026-01-09

```
# There is not one unique solution to this problem. Below is one example.

set.seed(30600)

# Generate one dataset under a given missingness mechanism.
simulate_data <- function(n, mechanism) {
  # Categorical covariate with unbalanced strata.
  x <- factor(sample(c("A", "B", "C", "D"), n, replace = TRUE,
                      prob = c(0.15, 0.25, 0.35, 0.25)))
  mu <- c(A = 0.25, B = 0.40, C = 0.60, D = 0.75)[x]
  phi <- 10
  y <- rbeta(n, mu * phi, (1 - mu) * phi)

  if (mechanism == "MCAR") {
    # Random half observed.
    idx <- sample.int(n, size = n / 2)
    r <- integer(n)
    r[idx] <- 1
  } else if (mechanism == "MAR") {
    # Missingness depends on X only (all strata have positive chance).
    p_mar <- c(A = 0.20, B = 0.40, C = 0.60, D = 0.80)[x]
    idx <- sample.int(n, size = n / 2, prob = p_mar)
    r <- integer(n)
    r[idx] <- 1
  } else if (mechanism == "MNAR") {
    # Missingness depends on Y (harder cutoff).
    score <- y + rnorm(n, sd = 0.05)
    r <- as.integer(score >= median(score))
  } else {
    stop("Unknown mechanism.")
  }

  list(x = x, y = y, r = r, mu = mean(y))
}

# Estimate Manski bounds and MCAR/MAR means.
estimate_all <- function(y, r, x) {
  lower <- mean(ifelse(r == 1, y, 0))
  upper <- mean(ifelse(r == 1, y, 1))
  mcar <- if (sum(r == 1) == 0) NA_real_ else mean(y[r == 1])

  x <- factor(x)
```

```

bin_means <- tapply(y[r == 1], x[r == 1], mean)
bin_weights <- prop.table(table(x))
mar <- sum(bin_means[names(bin_weights)] * bin_weights)

list(lower = lower, upper = upper, mcar = mcar, mar = mar)
}

# Run the simulation and return a summary table.
run_simulation <- function(S = 500, n = 200) {
  mechanisms <- c("MCAR", "MAR", "MNAR")
  rows <- list()

  for (mech in mechanisms) {
    mu_true <- numeric(S)
    lower <- numeric(S)
    upper <- numeric(S)
    mcar <- numeric(S)
    mar <- numeric(S)

    for (s in seq_len(S)) {
      dat <- simulate_data(n, mech)
      est <- estimate_all(dat$y, dat$r, dat$x)
      mu_true[s] <- dat$mu
      lower[s] <- est$lower
      upper[s] <- est$upper
      mcar[s] <- est$mcar
      mar[s] <- est$mar
    }

    rows[[length(rows) + 1]] <- data.frame(
      mechanism = mech,
      estimator = "MCAR",
      bias = mean(mcar - mu_true, na.rm = TRUE),
      variance = var(mcar, na.rm = TRUE)
    )
    rows[[length(rows) + 1]] <- data.frame(
      mechanism = mech,
      estimator = "MAR",
      bias = mean(mar - mu_true, na.rm = TRUE),
      variance = var(mar, na.rm = TRUE)
    )
    rows[[length(rows) + 1]] <- data.frame(
      mechanism = mech,
      estimator = "Manski bounds",
      manski_coverage = mean(lower <= mu_true & mu_true <= upper, na.rm = TRUE),
      avg_bounds_width = mean(upper - lower, na.rm = TRUE)
    )
  }
}

out <- list(do.call(rbind, rows[-c(3,6,9)]),
            do.call(rbind, rows[c(3,6,9)]))
out
}

```

```

# Run and print a clean summary table.
results <- run_simulation(S = 500, n = 200)

print(results, row.names = FALSE, right = FALSE)

## [[1]]
##   mechanism estimator bias      variance
##   MCAR        MCAR    -0.0008678540 0.0004983024
##   MCAR        MAR     -0.0009018292 0.0003478024
##   MAR         MCAR     0.0470909644 0.0004247283
##   MAR         MAR     -0.0003827734 0.0003705931
##   MNAR        MCAR     0.1832555368 0.0002646836
##   MNAR        MAR     0.1383490639 0.0003660825
##
## [[2]]
##   mechanism estimator manski_coverage avg_bounds_width
##   MCAR        Manski bounds 1           0.5
##   MAR         Manski bounds 1           0.5
##   MNAR        Manski bounds 1           0.5

```