

PS1 Solutions

Jingle Fu

Solution (a).

```
1 set.seed(2025)
2 n <- 100
3 u_i <- rnorm(n, mean = 0, sd = sqrt(5))
4 g_i <- rgamma(n, shape = 2, scale = 2)
5 r_i <- rbinom(n, size = 1, prob = 0.5)
6 x_star_i <- numeric(n)
7
8 for (i in 1:n) {
9   if (r_i[i] == 1) {
10    x_star_i[i] <- rgamma(1, shape = 3, scale = 1)
11   } else {
12     x_star_i[i] <- rgamma(1, shape = 7, scale = 1)
13   }
14 }
15
16 beta_0 <- 400
17 beta_1 <- 5
18 beta_2 <- 200
19 beta_3 <- 10
20
21 y_i <- beta_0 + beta_1 * x_star_i + beta_2 * r_i + beta_3 * g_i + u_i
22
23 n_i <- rnorm(n, mean = 10, sd = sqrt(3))
24 b_i <- rnorm(n, mean = 5 + sqrt(x_star_i), sd = sqrt(3))
25
26 data <- data.frame(
27   y = y_i,
28   x_star = x_star_i,
29   r = r_i,
30   g = g_i,
31   n = n_i,
32   b = b_i
33 )
```

Solution (b).

We consider the true model:

$$y_i = \beta_0 + \beta_1 x_i^* + \beta_2 r_i + \beta_3 g_i + u_i,$$

with the following Data Generating Process (DGP):

- $u_i \sim N(0, 5)$.
- $g_i \sim \Gamma(2, 2)$, so that

$$\mathbb{E}[g_i] = \frac{2}{2} = 1, \quad \mathbb{V}[g_i] = 2 \times 2^2 = 8.$$

- $r_i \in \{0, 1\}$ with $P(r_i = 1) = 0.5$.
- Conditionally on r_i , the fertilizer variable x_i^* is distributed as:

- If $r_i = 1$: $x_i^* \sim \Gamma(3, 1)$, so that

$$\mathbb{E}[x_i^* \mid r_i = 1] = 3, \quad \mathbb{V}[x_i^* \mid r_i = 1] = 3.$$

- If $r_i = 0$: $x_i^* \sim \Gamma(7, 1)$, so that

$$\mathbb{E}[x_i^* \mid r_i = 0] = 7, \quad \mathbb{V}[x_i^* \mid r_i = 0] = 7.$$

By the law of total expectation, we have:

$$\mathbb{E}[x_i^*] = 0.5 \cdot 3 + 0.5 \cdot 7 = 5.$$

Similarly, by the law of total variance:

$$\mathbb{V}[x_i^*] = \mathbb{E}[\mathbb{V}[x_i^* \mid r_i]] + \mathbb{V}(\mathbb{E}[x_i^* \mid r_i]) = 0.5 \cdot 3 + 0.5 \cdot 7 + 0.5[(3-5)^2 + (7-5)^2] = 5 + 4 = 9.$$

Regression 1: $y_i = \beta_0 + \beta_1 x_i^* + \text{error}_i$

The working regression is

$$y_i = \beta_0 + \beta_1 x_i^* + \varepsilon_i, \quad \text{with} \quad \varepsilon_i = \beta_2 r_i + \beta_3 g_i + u_i.$$

The OLS estimator for β_1 is given by

$$\hat{\beta}_1 = \beta_1 + \frac{\text{Cov}(x_i^*, \beta_2 r_i + \beta_3 g_i)}{\mathbb{V}[x_i^*]}.$$

By linearity,

$$\text{Cov}(x_i^*, \beta_2 r_i + \beta_3 g_i) = \beta_2 \text{Cov}(x_i^*, r_i) + \beta_3 \text{Cov}(x_i^*, g_i).$$

Calculation of $\text{Cov}(x_i^*, r_i)$:

$$\text{Cov}(x_i^*, r_i) = \mathbb{E}[x_i^* r_i] - \mathbb{E}[x_i^*] \mathbb{E}[r_i].$$

Since

$$\mathbb{E}[x_i^* r_i] = P(r_i = 1)\mathbb{E}[x_i^* | r_i = 1] + P(r_i = 0) \cdot 0 = 0.5 \cdot 3 = 1.5,$$

and $\mathbb{E}[r_i] = 0.5$, it follows that

$$\text{Cov}(x_i^*, r_i) = 1.5 - 5 \cdot 0.5 = 1.5 - 2.5 = -1.$$

Calculation of $\text{Cov}(x_i^*, g_i)$: From the construction of g_i , we have $\text{Cov}(x_i^*, g_i) = 0$.

Thus, the probability limit of $\hat{\beta}_1$ is:

$$\text{plim } \hat{\beta}_1 = \beta_1 + \frac{\beta_2(-1) + \beta_3(0)}{9} = \beta_1 - \frac{\beta_2}{9} \approx -17.22.$$

Our simulated result reports a mean of -16.981, with a standard error of 2.5386, which is close to our theoretical expectation, but far from the true value of 5.

Regression 2: $y_i = \beta_0 + \beta_1 x_i^* + \beta_2 r_i + \text{error}_i$

Now the regression is:

$$y_i = \beta_0 + \beta_1 x_i^* + \beta_2 r_i + \varepsilon_i, \quad \text{with} \quad \varepsilon_i = \beta_3 g_i + u_i.$$

By the Frisch–Waugh–Lovell theorem, we partial out r_i from x_i^* . Define the residual:

$$\tilde{x}_i = x_i^* - \mathbb{E}[x_i^* | r_i],$$

with

$$\mathbb{E}[x_i^* | r_i] = \begin{cases} 3, & r_i = 1, \\ 7, & r_i = 0. \end{cases}$$

Then the OLS estimator becomes:

$$\hat{\beta}_1 = \beta_1 + \frac{\text{Cov}(\tilde{x}_i, \beta_3 g_i)}{\mathbb{V}[\tilde{x}_i]} = \beta_1 + \beta_3 \frac{\text{Cov}(\tilde{x}_i, g_i)}{\mathbb{V}[\tilde{x}_i]}.$$

Assuming that $\mathbb{E}[g_i | r_i] = \mathbb{E}[g_i] = 1$, note that

$$\text{Cov}(\tilde{x}_i, g_i) = \text{Cov}(x_i^*, g_i) - \text{Cov}(\mathbb{E}[x_i^* | r_i], g_i).$$

A brief calculation shows:

$$\text{Cov}(\mathbb{E}[x_i^* | r_i], g_i) = 0.5 [3 \cdot 1 + 7 \cdot 1] - \mathbb{E}[x_i^*] \mathbb{E}[g_i] = 5 - 5 = 0.$$

Also,

$$\text{Cov}(x_i^*, g_i) = 0$$

Thus, the probability limit is:

$$\text{plim } \hat{\beta}_1^{(2)} = \beta_1.$$

Its variance is given by

$$\sigma_\epsilon^2 = \mathbb{V}[\beta_3 g_i + u_i] = \beta_3^2 \mathbb{V}[g_i] + 5.$$

Our simulated result reports a mean of 4.3648, with a standard error of 1.2873, which is closer to the true value and less volatile than regression (1).

Regression 3: $y_i = \beta_0 + \beta_1 x_i^* + \beta_2 r_i + \beta_3 g_i + \text{error}_i$

Here the regression exactly matches the true DGP:

$$y_i = \beta_0 + \beta_1 x_i^* + \beta_2 r_i + \beta_3 g_i + u_i.$$

Under the exogeneity assumption $\mathbb{E}[u_i \mid x_i^*, r_i, g_i] = 0$, the OLS estimator for β_1 is **consistent**:

$$\text{plim } \hat{\beta}_1^{(3)} = \beta_1.$$

Its finite-sample variance is given by the standard OLS formula:

$$\mathbb{V}[\hat{\beta}_1^{(3)}] = \sigma_\epsilon^2 [(X'X)^{-1}]_{11},$$

where the design matrix X includes the moments and cross-moments of x_i^* , r_i , and g_i .

The simulated result reports a mean of 5.0395, with a standard error of 0.0957, which is almost the true value.

Regression 4: $y_i = \beta_0 + \beta_1 x_i^* + \beta_2 r_i + \beta_4 n_i + \text{error}_i$

Since $n_i \sim N(10, 3)$ is generated independently of x_i^* , r_i , g_i , and u_i , it is an *irrelevant regressor*. Under the standard OLS assumptions, the inclusion of an irrelevant regressor does not cause bias in the estimated coefficient of x_i^* :

$$\text{plim } \hat{\beta}_1^{(4)} = \beta_1.$$

However, its inclusion may increase the finite-sample variance of $\hat{\beta}_1$. In particular, the variance formula now becomes

$$\mathbb{V}[\hat{\beta}_1^{(4)}] = \sigma_\epsilon^2 [(X'X)^{-1}]_{11},$$

where the design matrix X now includes the column corresponding to n_i . If n_i is only weakly correlated with x_i^* , then the increase in variance is modest. In summary, **Regression 4** yields a consistent estimator for β_1 , with no additional bias but possibly a slight inflation in variance.

The simulated result reports a mean of 4.4641, with a standard error of 1.3123, which satisfies our expectation. This is because the model introduces an uncorrelated variable

n_i that does not affect the estimation of β_1 , but reduced the degree of freedom in the model. So, it's variance is higher than regression (2).

Regression 5: $y_i = \beta_0 + \beta_1 x_i^* + \beta_2 r_i + \beta_4 b_i + \text{error}_i$

Here,

$$b_i \sim N\left(5 + \sqrt{x_i^*}, 3\right),$$

so that b_i is a non-linear function of x_i^* . This implies that b_i is correlated with x_i^* . In particular, since

$$\mathbb{E}[b_i | x_i^*] = 5 + \sqrt{x_i^*},$$

we have

$$\text{Cov}(x_i^*, b_i) \neq 0.$$

The inclusion of b_i does not cause endogeneity provided that

$$\mathbb{E}[u_i | x_i^*, r_i, g_i, b_i] = 0.$$

Thus, by the Frisch–Waugh–Lovell theorem, the coefficient β_1 is still identified and

$$\text{plim } \hat{\beta}_1^{(5)} = \beta_1.$$

However, the strong correlation between x_i^* and b_i increases multicollinearity. To see this more formally, consider the variance of $\hat{\beta}_1$ in a multiple regression:

$$\mathbb{V}[\hat{\beta}_1^{(5)}] = \sigma_\epsilon^2 [(X'X)^{-1}]_{11}.$$

When x_i^* is highly collinear with b_i , the effective variation in x_i^* (after partialling out the effect of b_i along with r_i and g_i) is reduced. Denote by $R_{x,b}^2$ the coefficient of determination from regressing x_i^* on the other regressors (including b_i). Then, the variance inflation factor (VIF) for $\hat{\beta}_1$ is given by

$$\text{VIF} = \frac{1}{1 - R_{x,b}^2}.$$

Thus, the asymptotic variance becomes

$$\mathbb{V}[\hat{\beta}_1^{(5)}] \approx \frac{\sigma_\epsilon^2}{n \mathbb{V}[x_i^*]} \cdot \frac{1}{1 - R_{x,b}^2},$$

which is larger than that in Regression 3. In summary, while **Regression 5** still provides a consistent estimate of β_1 , the estimator's variance is inflated due to the high collinearity between x_i^* and b_i .

The simulated result reports a mean of 4.5369, with a standard error of 1.3026, which estimated β worse than regression (4), with a higher standard error than regression (2). Also, it's variance is higher than regression (2)

```
1 reg1 <- lm(y ~ x_star, data = data)
```

```

2 summary_reg1 <- summary(reg1)
3
4 reg2 <- lm(y ~ x_star + r, data = data)
5 summary_reg2 <- summary(reg2)
6
7 reg3 <- lm(y ~ x_star + r + g, data = data)
8 summary_reg3 <- summary(reg3)
9
10 reg4 <- lm(y ~ x_star + r + n, data = data)
11 summary_reg4 <- summary(reg4)
12
13 reg5 <- lm(y ~ x_star + r + b, data = data)
14 summary_reg5 <- summary(reg5)
15
16 extract_results <- function(reg_summary, reg_name) {
17   beta1_estimate <- reg_summary$coefficients["x_star", "Estimate"]
18   beta1_se <- reg_summary$coefficients["x_star", "Std. Error"]
19   beta1_true <- 5
20
21   cat(paste0("\n", reg_name, ":\n"))
22   cat(paste0("Estimated beta_1: ", round(beta1_estimate, 4), "\n"))
23   cat(paste0("True beta_1: ", beta1_true, "\n"))
24   cat(paste0("Standard Error: ", round(beta1_se, 4), "\n"))
25   cat(paste0("Difference from true value: ", round(beta1_estimate -
26     beta1_true, 4), "\n"))
27   cat(paste0("Adjusted R_2: ", round(reg_summary$adj.r.squared, 4), "\n"))
28 }
29
30 extract_results(summary_reg1, "Regression 1 (y ~ x_star)")
31 extract_results(summary_reg2, "Regression 2 (y ~ x_star + r)")
32 extract_results(summary_reg3, "Regression 3 (y ~ x_star + r + g)")
33 extract_results(summary_reg4, "Regression 4 (y ~ x_star + r + n)")
34 extract_results(summary_reg5, "Regression 5 (y ~ x_star + r + b)")

```

Table 1: Regression Results for Question (b)

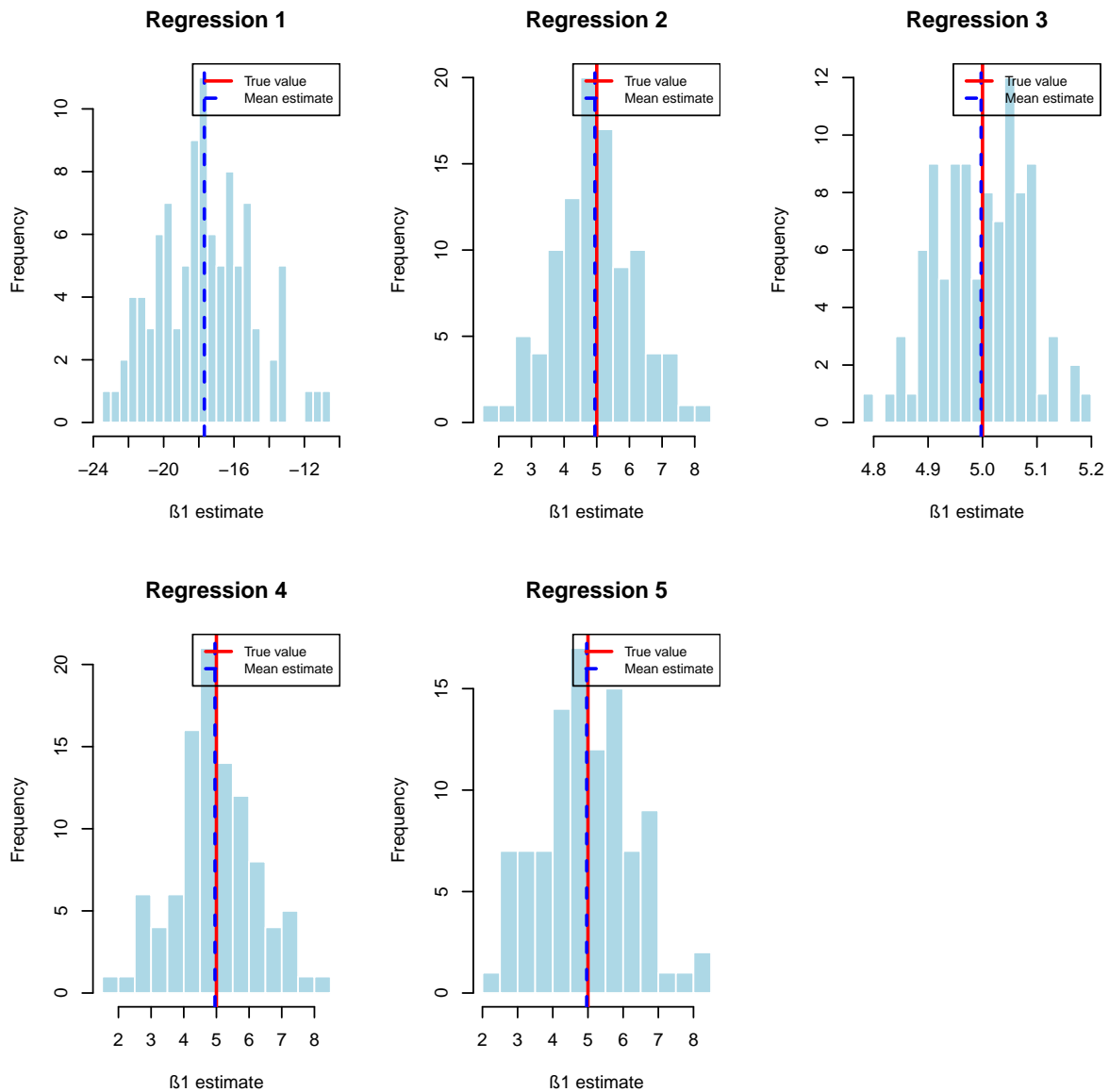
Regression	$\hat{\beta}_1$	SE	Difference	Adjusted R^2
$y \sim x^*$	-16.9809	2.5386	-21.9809	0.3065
$y \sim x^* + r$	4.3648	1.2873	-0.6352	0.9029
$y \sim x^* + r + g$	5.0395	0.0957	0.0395	0.9995
$y \sim x^* + r + n$	4.4641	1.3123	-0.5359	0.9021
$y \sim x^* + r + b$	4.5369	1.3026	-0.4631	0.9027

Solution (c).

From a theoretical perspective, the Monte Carlo simulation allows us to empirically approximate the sampling distribution of each estimator. The histograms should confirm

our theoretical derivations:

1. $\hat{\beta}_1^{(1)}$ should show a systematic bias from the true value of 5.
2. $\hat{\beta}_1^{(2)}$ might show a smaller bias if there's still correlation between g_i and x_i^* after controlling for r_i .
3. $\hat{\beta}_1^{(3)}$, $\hat{\beta}_1^{(4)}$, and $\hat{\beta}_1^{(5)}$ should be centered around 5, but with increasing variance.



```

1 set.seed(2025)
2 M <- 100
3 n <- 100
4
5 beta1_estimates <- matrix(NA, nrow = M, ncol = 5)

```

```

6 colnames(beta1_estimates) <- c("Reg1", "Reg2", "Reg3", "Reg4", "Reg5")
7
8 for (m in 1:M) {
9   u_i <- rnorm(n, mean = 0, sd = sqrt(5))
10  g_i <- rgamma(n, shape = 2, scale = 2)
11  r_i <- rbinom(n, size = 1, prob = 0.5)
12
13  x_star_i <- numeric(n)
14  for (i in 1:n) {
15    if (r_i[i] == 1) {
16      x_star_i[i] <- rgamma(1, shape = 3, scale = 1)
17    } else {
18      x_star_i[i] <- rgamma(1, shape = 7, scale = 1)
19    }
20  }
21
22  beta_0 <- 400
23  beta_1 <- 5
24  beta_2 <- 200
25  beta_3 <- 10
26
27  y_i <- beta_0 + beta_1 * x_star_i + beta_2 * r_i + beta_3 * g_i + u_i
28
29  n_i <- rnorm(n, mean = 10, sd = sqrt(3))
30  b_i <- rnorm(n, mean = 5 + sqrt(x_star_i), sd = sqrt(3))
31
32  data <- data.frame(
33    y = y_i,
34    x_star = x_star_i,
35    r = r_i,
36    g = g_i,
37    n = n_i,
38    b = b_i
39  )
40
41  reg1 <- lm(y ~ x_star, data = data)
42  reg2 <- lm(y ~ x_star + r, data = data)
43  reg3 <- lm(y ~ x_star + r + g, data = data)
44  reg4 <- lm(y ~ x_star + r + n, data = data)
45  reg5 <- lm(y ~ x_star + r + b, data = data)
46
47  # Store estimates
48  beta1_estimates[m, 1] <- coef(reg1)["x_star"]
49  beta1_estimates[m, 2] <- coef(reg2)["x_star"]
50  beta1_estimates[m, 3] <- coef(reg3)["x_star"]
51  beta1_estimates[m, 4] <- coef(reg4)["x_star"]
52  beta1_estimates[m, 5] <- coef(reg5)["x_star"]
53 }

```



```

54
55 beta1_df <- data.frame(
56   Estimate = c(beta1_estimates),
57   Regression = rep(colnames(beta1_estimates), each = M)
58 )
59
60 beta1_summary <- data.frame(
61   Regression = colnames(beta1_estimates),
62   Mean = colMeans(beta1_estimates),
63   SE = apply(beta1_estimates, 2, sd),
64   Bias = colMeans(beta1_estimates) - 5
65 )
66
67 par(mfrow = c(2, 3))
68 for (i in 1:5) {
69   hist(beta1_estimates[, i],
70        main = paste0("Regression", i),
71        xlab = "estimate",
72        breaks = 20,
73        col = "lightblue",
74        border = "white")
75   abline(v = 5, col = "red", lwd = 2) # True value
76   abline(v = mean(beta1_estimates[, i]), col = "blue", lty = 2,
77          lwd = 2) # Mean estimate

```

Solution (d).

When $x_i^* \mid (r_i = 1) = x_i^* \mid (r_i = 0) \sim \Gamma(5, 1)$

If we set

$$x_i^* \mid (r_i = 1) = x_i^* \mid (r_i = 0) \sim \Gamma(5, 1),$$

then

$$\mathbb{E}[x_i^* \mid r_i] = 5 \quad \text{for both } r_i = 0, 1,$$

and hence

$$\mathbb{E}[x_i^*] = 5 \quad \text{and} \quad \mathbb{V}[x_i^*] = 5.$$

In this case,

$$\text{Cov}(x_i^*, r_i) = \mathbb{E}[x_i^* r_i] - \mathbb{E}[x_i^*] \mathbb{E}[r_i] = 0.5 \cdot 5 - 5 \cdot 0.5 = 0.$$

Thus, in Regression 1 the omitted variable bias reduces to:

$$\text{Bias}^{(1)} = \frac{0 + \beta_3 \text{Cov}(x_i^*, g_i)}{5} = 0.$$

Regressions (2) and (3) remain unbiased. Although Regression (1) is now unbiased, omitting r_i may still inflate the variance due to residual variation in y_i .

The relationship of variance remains the same: $\mathbb{V}[\hat{\beta}_1^{(3)}] < \mathbb{V}[\hat{\beta}_1^{(2)}] < \mathbb{V}[\hat{\beta}_1^{(1)}]$.

When $\beta_2 = 0$

Then the bias in Regression 1 simplifies to:

$$\text{Bias}^{(1)} = \frac{\beta_3 \text{Cov}(x_i^*, g_i)}{\mathbb{V}[x_i^*]} = 0.$$

Hence, the estimator is unbiased in Regression (1) even if r_i is omitted.

Regressions (2) and (3) are likewise unbiased; however, including r_i now only increases the number of regressors without providing explanatory power, potentially affecting efficiency.

The relationship of variance change: $\mathbb{V}[\hat{\beta}_1^{(3)}] < \mathbb{V}[\hat{\beta}_1^{(1)}] < \mathbb{V}[\hat{\beta}_1^{(2)}]$.

When $r_i = 1$ with probability 0.1

Then,

$$\mathbb{E}[r_i] = 0.1, \quad \mathbb{E}[x_i^*] = 0.1 \cdot 3 + 0.9 \cdot 7 = 6.6,$$

and

$$\mathbb{E}[x_i^* r_i] = 0.1 \cdot 3 = 0.3.$$

Thus,

$$\text{Cov}(x_i^*, r_i) = 0.3 - 6.6 \cdot 0.1 = 0.3 - 0.66 = -0.36.$$

Accordingly, the bias in Regression 1 becomes:

$$\text{Bias}^{(1)} = \frac{\beta_2(-0.36) + \beta_3 \text{Cov}(x_i^*, g_i)}{\mathbb{V}[x_i^*]} = -\frac{0.36\beta_2}{\mathbb{V}[x_i^*]},$$

with $\mathbb{V}[x_i^*]$ recalculated under the new mixture proportions. This is smaller in magnitude than in the balanced case (where $P(r_i = 1) = 0.5$). Regressions (2) and (3) remain unbiased.

The relationship of variance remains the same: $\mathbb{V}[\hat{\beta}_1^{(3)}] < \mathbb{V}[\hat{\beta}_1^{(2)}] < \mathbb{V}[\hat{\beta}_1^{(1)}]$.

When $\beta_3 = 50$

In Regression (1), the omitted error is

$$\varepsilon_i = \beta_2 r_i + 50 g_i + u_i.$$

Since $\text{Cov}(x_i^*, g_i) = 0$, the bias in Regression (1) remains

$$\text{Bias}^{(1)} = \beta_2 \frac{\text{Cov}(x_i^*, r_i)}{\text{Var}(x_i^*)}.$$

However, the variance of the error term increases dramatically due to the larger coefficient on g_i , thereby increasing the variance of $\hat{\beta}_1^{(1)}$.

In Regression (2), the error term becomes

$$\varepsilon_i = 50 g_i + u_i,$$

and hence $\hat{\beta}_1^{(2)}$ remains unbiased but less efficient.

Regression (3) includes g_i , so

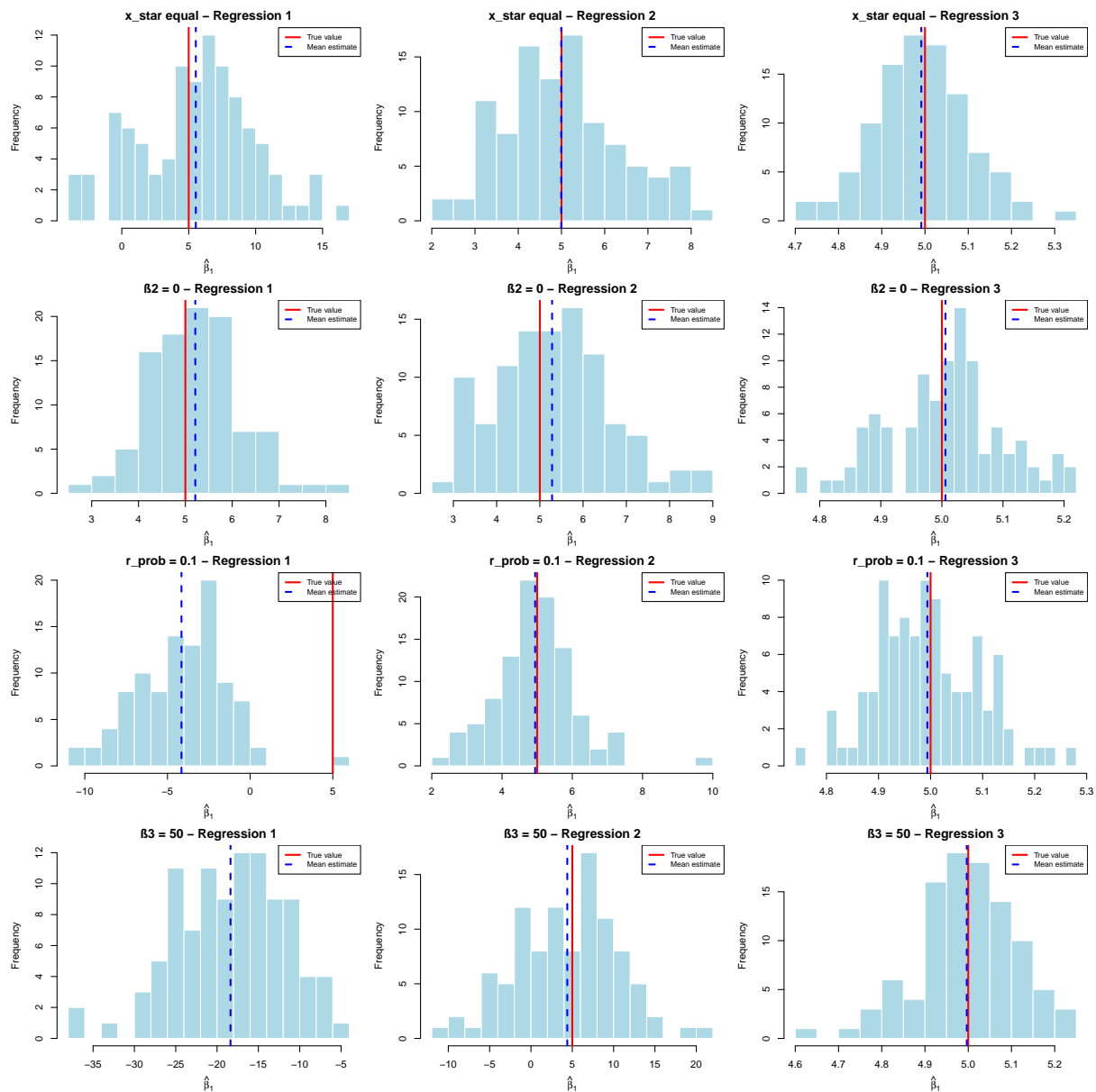
$$\mathbb{E}[\hat{\beta}_1^{(3)}] = \beta_1,$$

and this regression remains efficient despite the larger variability introduced by $50 g_i$.

The relationship of variance remains the same: $\mathbb{V}[\hat{\beta}_1^{(3)}] < \mathbb{V}[\hat{\beta}_1^{(2)}] < \mathbb{V}[\hat{\beta}_1^{(1)}]$.

Table 2: Simulation Summary Statistics for Question (d)

Scenario	Regression	Mean	SE	Bias
$x_i^* r_i = 1 = x_i^* r_i = 0$	Reg1	5.530295	4.3563060	0.5302951
$x_i^* r_i = 1 = x_i^* r_i = 0$	Reg2	4.994777	1.3270007	-0.0052234
$x_i^* r_i = 1 = x_i^* r_i = 0$	Reg3	4.991522	0.1105130	-0.0084783
$\beta_2 = 0$	Reg1	5.211875	0.9578191	0.2118745
$\beta_2 = 0$	Reg2	5.282412	1.2969897	0.2824116
$\beta_2 = 0$	Reg3	5.005810	0.0950364	0.0058101
$p(r_i = 1) = 0.1$	Reg1	-4.152274	2.6530512	-9.1522744
$p(r_i = 1) = 0.1$	Reg2	4.941895	1.1553780	-0.0581046
$p(r_i = 1) = 0.1$	Reg3	4.993826	0.0973356	-0.0061737
$\beta_3 = 50$	Reg1	-18.371055	6.5915167	-23.3710551
$\beta_3 = 50$	Reg2	4.394012	6.1969404	-0.6059882
$\beta_3 = 50$	Reg3	4.996611	0.1112587	-0.0033893



```

1 run_monte_carlo <- function(x_star_equal = FALSE, beta2_zero = FALSE, r_
  prob = 0.5, beta3_value = 10) {
2   M <- 100
3   n <- 100
4
5   beta1_estimates <- matrix(NA, nrow = M, ncol = 3)
6   colnames(beta1_estimates) <- c("Reg1", "Reg2", "Reg3")
7
8   for (m in 1:M) {
9     # Generate data according to modified DGP
10    u_i <- rnorm(n, mean = 0, sd = 5)
11    g_i <- rgamma(n, shape = 2, scale = 2)
12    r_i <- rbinom(n, size = 1, prob = r_prob)
13
14    x_star_i <- numeric(n)
15    if (x_star_equal) {

```

```

16     x_star_i <- rgamma(n, shape = 5, scale = 1)
17   } else {
18     for (i in 1:n) {
19       if (r_i[i] == 1) {
20         x_star_i[i] <- rgamma(1, shape = 3, scale = 1)
21       } else {
22         x_star_i[i] <- rgamma(1, shape = 7, scale = 1)
23       }
24     }
25   }
26
27   beta_0 <- 400
28   beta_1 <- 5
29   beta_2 <- ifels\mathbb{E}[beta2_zero, 0, 200]
30   beta_3 <- beta3_value
31
32   y_i <- beta_0 + beta_1 * x_star_i + beta_2 * r_i + beta_3 * g_i + u_
33   i
34
35   data <- data.frame(
36     y = y_i,
37     x_star = x_star_i,
38     r = r_i,
39     g = g_i
40   )
41
42   reg1 <- lm(y ~ x_star, data = data)
43   reg2 <- lm(y ~ x_star + r, data = data)
44   reg3 <- lm(y ~ x_star + r + g, data = data)
45
46   beta1_estimates[m, 1] <- coef(reg1)["x_star"]
47   beta1_estimates[m, 2] <- coef(reg2)["x_star"]
48   beta1_estimates[m, 3] <- coef(reg3)["x_star"]
49 }
50
51 return(beta1_estimates)
52 }
53
54 results_original <- run_monte_carlo()
55 results_xstar_equal <- run_monte_carlo(x_star_equal = TRUE)
56 results_beta2_zero <- run_monte_carlo(beta2_zero = TRUE)
57 results_r_prob_0.1 <- run_monte_carlo(r_prob = 0.1)
58 results_beta3_50 <- run_monte_carlo(beta3_value = 50)
59
60 calc_summary <- function(results, scenario_name) {
61   summary_df <- data.frame(
62     Scenario = rep(scenario_name, 3),
63     Regression = c("Reg1", "Reg2", "Reg3"),

```

```

63   Mean = colMeans(results),
64   SD = apply(results, 2, sd),
65   Bias = colMeans(results) - 5
66 )
67 return(summary_df)
68 }
69
70 summary_original <- calc_summary(results_original, "Original DGP")
71 summary_xstar_equal <- calc_summary(results_xstar_equal, "x_star equal")
72 summary_beta2_zero <- calc_summary(results_beta2_zero, " = 0")
73 summary_r_prob_0.1 <- calc_summary(results_r_prob_0.1, "r_prob = 0.1")
74 summary_beta3_50 <- calc_summary(results_beta3_50, " = 50")
75
76 all_summaries <- rbind(
77   summary_original,
78   summary_xstar_equal,
79   summary_beta2_zero,
80   summary_r_prob_0.1,
81   summary_beta3_50
82 )
83
84 latex_table_d <- kable(all_summaries, format = "latex", booktabs = TRUE,
85   caption = "Simulation Summary Statistics for
86   Question (d)")
87
88 output_file_d <- "d.tex"
89 cat(latex_table_d, file = output_file_d)
90
91 cat("\n%% Table saved to ", output_file_d, "\n", sep = "", file = output
92   _file_d, append = TRUE)
93
94 scenarios <- list(
95   "x_star equal" = results_xstar_equal,
96   " = 0" = results_beta2_zero,
97   "r_prob = 0.1" = results_r_prob_0.1,
98   " = 50" = results_beta3_50
99 )
100
101 pdf("Q(d).pdf", width = 12, height = 12)
102 par(mfrow = c(4, 3), mar = c(4, 4, 2, 1))
103 for (scenario in names(scenarios)) {
104   current <- scenarios[[scenario]]
105   for (i in 1:3) {
106     hist(current[, i],
107       main = paste(scenario, "- Regression", i),
108       xlab = expression(hat(beta)[1]),
109       breaks = 20,
110       col = "lightblue",
111       border = "white")

```

```
109     abline(v = 5, col = "red", lwd = 2)
110     abline(v = mean(current[, i]), col = "blue", lty = 2, lwd = 2)
111     legend("topright",
112           legend = c("True value", "Mean estimate"),
113           col = c("red", "blue"),
114           lty = c(1, 2),
115           lwd = 2,
116           cex = 0.8)
117   }
118 }
119 dev.off()
```