Final Solutions

Hanning Luo

May 7, 2025

Final Exam Instructions:

- This final exam is due on **May 5, 11:59 pm** Eastern time. Please upload a PDF of your solutions to **Gradescope**. When submitting, please match your responses with the questions.
- We will accept hand-written solutions but we strongly advise graduate students to typeset your answers in LaTeX. This is a semi-closed book test. You are **NOT** allowed to: search internet / AI for solutions or communicate amongst each other.
- You are allowed to utilize class materials (slides, section slides, pset solutions).
- Always use CGIS Knafel Zipcode **02138** as your seed for coding tasks.

1. OLS (30pt)

Suppose you are studying the relationship between the number of hours a student studies (X_i) and their exam score (Y_i). Assume the following data-generating process for each student $i = 1, \ldots, n$:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

where:

- $X_i \sim \text{Uniform}(0, 10)$, independent across i
- $\varepsilon_i \sim \text{Normal}(0, \sigma^2)$, independent of X_i and independent across i
- $\beta_0 = 50$, $\beta_1 = 5$, and $\sigma^2 = 16$.

Answer the following:

a. (6 points) Find the mean and variance of Y_i .

b. (6 points) By the Law of Large Numbers (LLN), what happens to the sample mean $\bar{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i$ as $n \to \infty$?

c. (6 points) By the Central Limit Theorem (CLT), approximate the distribution of $\sqrt{n}(\bar{Y}_n - \mathbb{E}[Y_i])$ as n becomes large.

d. (6 points) Suppose you run an OLS regression of Y_i on X_i (with intercept). What are the probability limits (i.e., plim, convergence in probability) of the OLS estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ as $n \to \infty$?

e. (6 points) Briefly explain why $\hat{\beta}_1$ is consistent under the given conditions (no need to formally prove, just check the necessary assumptions). What would happen if ε_i were correlated with X_i?

Solutions:

a. We have:

$$\mathbb{E}[Y_i] = \mathbb{E}[\beta_0 + \beta_1 X_i + \varepsilon_i] = \beta_0 + \beta_1 \mathbb{E}[X_i] + \mathbb{E}[\varepsilon_i]$$

Since $\mathbb{E}[X_i] = \frac{0+10}{2} = 5$ and $\mathbb{E}[\varepsilon_i] = 0$, we have:

$$\mathbb{E}[Y_i] = 50 + 5 \times 5 = 75$$

Variance:

$$\text{Var}(Y_i) = \text{Var}(\beta_0 + \beta_1 X_i + \varepsilon_i) = \beta_1^2 \text{Var}(X_i) + \text{Var}(\varepsilon_i)$$

For $X_i \sim \text{Uniform}(0, 10)$:

$$\text{Var}(X_i) = \frac{(10-0)^2}{12} = \frac{100}{12} = \frac{25}{3}$$

Thus:

$$\text{Var}(Y_i) = 5^2 \times \frac{25}{3} + 16 = \frac{625}{3} + 16 = \frac{625+48}{3} = \frac{673}{3} \approx 224.33$$

b. By the Law of Large Numbers (LLN), as $n \to \infty$, we have:

$$\bar{Y}_n \xrightarrow{p} \mathbb{E}[Y_i] = 75$$

c. By the Central Limit Theorem (CLT), as n becomes large:

$$\sqrt{n}(\bar{Y}_n - \mathbb{E}[Y_i]) \xrightarrow{d} \mathcal{N}(0, \text{Var}(Y_i))$$

That is:

$$\sqrt{n}(\bar{Y}_n - 75) \xrightarrow{d} \mathcal{N}\left(0, \frac{673}{3}\right)$$

d. Under the given assumptions (linear model correctly specified, X_i and ε_i independent, etc.), OLS estimators are consistent. Thus:

$$\text{plim}(\hat{\beta}_0) = \beta_0 = 50, \quad \text{plim}(\hat{\beta}_1) = \beta_1 = 5$$

e. $\hat{\beta}_1$ is consistent because:

- The model is correctly specified,
- ε_i is independent of X_i (exogeneity condition),
- $\text{Var}(X_i) > 0$ (no perfect multicollinearity). If ε_i were correlated with X_i, then the exogeneity assumption would be violated, and $\hat{\beta}_1$ would be biased and inconsistent, even as $n \to \infty$.

2. Publication Bias (30pt)

Political science journals rarely publish statistically insignificant results. Does this publication bias lead to systematic bias in our understanding of political phenomena? Let's use simulations to find out. We will consider the effect of three different political phenomena—in each case, we are interested in the effect of X on Y. The following are the "true'' models describing the relationship between each (X, Y) pair:

- $X_{1i} \sim N(0,1); \quad Y_{1i} = 2 + 0.1X_{1i} + u_{1i}; \quad u_{1i} \sim N(0,1)$
- $X_{2i} \sim N(0,1); \quad Y_{2i} = 2 + 5X_{2i} + u_{2i}; \quad u_{2i} \sim N(0,1)$
- $X_{3i} \sim N(0,1); \quad Y_{3i} = 2 + 0X_{3i} + u_{3i}; \quad u_{3i} \sim N(0,1)$

Thus, the only difference is the size of the true effect of X on Y. In the first model, X_1 has a very weak effect on Y_1; in the second, X_2 has a very strong effect on Y_2; and in the third, X_3 has no effect on Y_3.

a. (10pt) For each of the three phenomena, simulate 10,000 datasets of size 30 and calculate an OLS slope estimate for each. For each regression, record the difference between the estimated slope coefficient and the true value of β (i.e., the estimation error) and also record the p-value from the regression (you will use the p-value in the next part of this question). Plot the distribution (by histogram or density plot) of the *estimation error* among published articles for each of the three phenomena. Indicate the mean estimation error (an estimate for the bias) with a vertical line, and briefly interpret your plots in terms of bias.

Hint: To get a p-value from the model, first you run a linear regression: `reg1 <- lm(y ~ X)`. Then the p-value is obtained by `summary(reg1)$coefficients[2,4]`.

b. (10pt) Now consider a journal editorial policy such that empirical research is not published unless the results are statistically significant, meaning that the p-value on the coefficient of interest is ≤ 0.05. Using your simulations from part (a), make a density plot for the difference between the estimated coefficients and the true value for all the *publishable* results under this policy. Plot each of the three phenomena separately. What percentage of studies are considered publishable for each (X, Y) pairing? Do these studies correctly estimate the regression coefficient on average? Provide an intuitive explanation of the results. Should we be concerned by the findings?

c. (10pt) In the previous part, we assumed that only studies with significant results ended up getting published, but of course that is not realistic—null results are sometimes published. Let us assume instead that the journal is still willing to publish some null results, but significant results are much more likely to be published. Recreate your plots from part (b), but this time selecting articles for publication according to Bern(p_i) where $p_i = 0.95$ if the study has significant results and $p_i = 0.05$ if the study does not. Separately for each phenomenon, sample a total of $n = 1000$ studies from the 10,000 you generated in part (a). Make sure to plot a vertical line at the mean of each of your plots to represent the bias. What do you notice? Under which phenomena should we be worried about this publishing practice?

Hint: Recall weighted sampling in R from section zero, you can first generate a vector `prob_vec` with each unit's probability of being sampled, and then apply `sample(units, size = n, prob = prob_vec)`

Solutions:

```r
set.seed(02138)

# Function to simulate one dataset and run regression
simulate_one <- function(beta, n = 30) {
  X <- rnorm(n)
  u <- rnorm(n)
  Y <- 2 + beta * X + u
  model <- lm(Y ~ X)
  coef_est <- coef(model)[2]
  p_val <- summary(model)$coefficients[2, 4]
  est_error <- coef_est - beta
  return(c(est_error = est_error, p_val = p_val))
}
```

```r
# Set parameters
n_sim <- 10000
sample_size <- 30
true_betas <- c(0.1, 5, 0)

# Storage
results_list <- list()

# Simulate for each phenomenon
for (i in 1:3) {
  beta <- true_betas[i]
  res <- replicate(n_sim, simulate_one(beta, sample_size))
  rownames(res) <- c("est_error", "p_val")
  res_df <- data.frame(
    est_error = res["est_error", ],
    p_value = res["p_val", ]
  )
  results_list[[i]] <- res_df
}

names(results_list) <- c("Weak Effect", "Strong Effect", "Null Effect")

# Plot
pdf("./Downloads/a.pdf", width = 9, height = 4)
par(mfrow = c(1, 3))  # 3 plots side-by-side

for (i in 1:3) {
  res_df <- results_list[[i]]
  hist(res_df$est_error, breaks = 50, main = names(results_list)[i],
       xlab = "Estimation Error", probability = TRUE,xlim=c(-1,0.6),
       col = "lightgray", border = "white")
  abline(v = mean(res_df$est_error), col = "red", lwd = 2)
  legend("topleft", legend = paste0("Mean Bias: \n",
                                    round(mean(res_df$est_error), 4)),
         col = "red", bty = "n")
}
dev.off()
```

```
## pdf
##   2
```

```r
############## b
# Base R plotting
pdf("./Downloads/b.pdf", width = 9, height = 4)
par(mfrow = c(1, 3))  # 3 plots side-by-side

publishable_pct <- numeric(3)  # To store percentages

for (i in 1:3) {
  res_df <- results_list[[i]]

  # Filter to publishable studies (p <= 0.05)
  publishable <- res_df[res_df$p_value <= 0.05, ]
```

```r
  # Calculate publishable percentage
  publishable_pct[i] <- nrow(publishable) / nrow(res_df) * 100

  # Plot density of estimation error
  plot(density(publishable$est_error), xlim = c(-1, 1),
       main = paste0(names(results_list)[i], "\n",
                     round(publishable_pct[i], 1), "% Publishable"),
       xlab = "Estimation Error", lwd = 2)

  # Add vertical line at mean bias
  abline(v = mean(publishable$est_error), col = "red", lwd = 2)

  # Add mean bias to plot
  legend("topleft",
         legend = paste0("Mean \n Published \n Bias: \n ", round(mean(publishable$est_error), 4)),
         col = "red", bty = "n")
}
dev.off()
```

```
## pdf
##   2
```

```r
####### c
set.seed(02138)

n_sample <- 1000  # Number of studies to select after publication

# Base R plotting
pdf("./Downloads/c.pdf", width = 9, height = 4)
par(mfrow = c(1, 3))  # 3 plots side-by-side

# Storage for sampling percentages
sampling_info <- list()

for (i in 1:3) {
  res_df <- results_list[[i]]

  # Create publication probability
  pub_prob <- ifelse(res_df$p_value <= 0.05, 0.95, 0.05)

  # Randomly select studies based on Bernoulli(pub_prob)
  published_flag <- rbinom(n = nrow(res_df), size = 1, prob = pub_prob)
  published_studies <- res_df[published_flag == 1, ]

  # Sample exactly 1000 studies (if more than 1000 available)
  if (nrow(published_studies) >= n_sample) {
    published_sample <- published_studies[sample(1:nrow(published_studies), n_sample), ]
  } else {
    warning(paste("Not enough published studies for", names(results_list)[i]))
    published_sample <- published_studies
  }

  sampling_info[[i]] <- nrow(published_studies) / nrow(res_df) * 100  # Save % published
```

```
  # Plot density
  plot(density(published_sample$est_error), xlim = c(-1, 1),
       main = paste0(names(results_list)[i], "\nSampled ", nrow(published_sample), " studies"),
       xlab = "Estimation Error", lwd = 2)

  # Add vertical line at mean bias
  abline(v = mean(published_sample$est_error), col = "red", lwd = 2)

  # Add mean bias to plot
  legend("topleft",
         legend = paste0("Mean Bias: \n", round(mean(published_sample$est_error), 4)),
         col = "red",  bty = "n")
}
```

```
## Warning: Not enough published studies for Null Effect
```

```
dev.off()
```

```
## pdf
##   2
```

a. For each plot, the estimation error distribution tightly clusters around zero. OLS provides unbiased estimates of the true effect.

b. All studies with strong effect are publishable. Most studies with weak or no effect are not publishable, but a few others show significant results due to the random noise term. Therefore, even when the true effect is negligible, there could still be false discovery.

c. Probabilistic publication reduces the damage of publication bias. But small effects remain at risk of exaggeration: the estimation bias is not eliminated.

3. Asymptotic Normality and Missing Data (40 pt)

You will conduct a simulation study to explore the impact of missing data on confidence intervals and estimator behavior. The true data-generating process is:

- $X_1 \sim \mathcal{N}(-4, 0.5)$
- $X_2 = 0.5X_1 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$
- $Y = 1 + 2 \cdot X_1 - 1 \cdot X_2 + \eta$, where $\eta \sim \mathcal{N}(0, 1)$

Missingness is introduced in X_1 according to:

$$\Pr(X_1 \text{ missing}) = \text{logit}^{-1}(X_2 + e)$$

where $e \sim N(1, 1)$, and

$$\text{logit}^{-1}(x) = \frac{1}{1 + e^{-x}}$$

a. (5pt) Explain why the missing data mechanism is Missing At Random (MAR).

b. (5pt) Write down the definition of asymptotic normality. What does it mean when a confidence interval has nominal coverage?

c. (10pt) Simulate 1000 datasets with $n = 500$ observations each. For each simulated dataset:

- Estimate coefficients using (i) Oracle data (no missingness), (ii) Complete case analysis, and (iii) Multiple Imputation (5 imputations).
- Store point estimates and standard errors for β_{X_1} and β_{X_2}.
- You can use the R package `mice` for multiple imputation.

For each method and each coefficient, calculate the empirical coverage probability of confidence intervals at levels from 1% to 99%. Also, Write down *analytically* the general formula you use for any given level of confidence interval.

Hint: To generate missingness, you can first generate a `prob_vec` to store the probability of missing X_1 for each unit (you can check out the `plogis` function), and then generate a vector to indicate each unit's missingness by, for example, `miss_indicator <- rbinom(n, size = 1, prob = prob_vec)`.

d. (10pt) For each method, plot the empirical coverage curves (y-axis) against nominal confidence levels (x-axis, from 1% to 99%). Discuss:

- Whether the estimators are approximately unbiased.
- Whether the nominal confidence level matches the empirical coverage.

e. (10pt) Which method (Oracle, Complete Case, MI) appears most reliable under this missing data mechanism? Provide an explanation based on your findings. You can use more visualization to support your claims (for example, you can plot the distribution of simulated estimates under each method).

Solutions:

a. The missingness in X_1 depends on an observed variable X_2, not on unobserved values of X_1 or the outcome Y, so it is MAR.

b. Asymptotic normality means as the sample size $n \to \infty$, the estimator $\hat{\theta}_n$ becomes normally distributed around the true value θ. A confidence interval has nominal coverage if its empirical coverage (the fraction of times it contains the true value in repeated samples) matches the stated confidence level.

c. Formula for α level CI coverage:

$$[\hat{\beta} - t \times \hat{\text{se}}, \hat{\beta} + t \times \hat{\text{se}}]$$

where t is the $1 - (1 - \alpha)/2$th quantile of $t(n - 3)$ distribution.

```r
library(mice)

# Storage
n_sim <- 500
n <- 1000

# Each row = one simulation, columns = estimates and standard errors
results_oracle <- matrix(NA, nrow = n_sim, ncol = 4)
results_complete <- matrix(NA, nrow = n_sim, ncol = 4)
results_mi <- matrix(NA, nrow = n_sim, ncol = 4)
```

```r
colnames(results_oracle) <- c("beta_X1", "beta_X2", "se_X1", "se_X2")
colnames(results_complete) <- colnames(results_oracle)
colnames(results_mi) <- colnames(results_oracle)

set.seed(02138)

for (sim in 1:n_sim) {

  # 1. Simulate data
  X1 <- rnorm(n, mean = -4, sd = 0.5)
  X2 <- 0.5*X1 + rnorm(n, mean = 0, sd = 1)
  Y <- 1 + 2 * X1 - 1 * X2 + rnorm(n, mean = 0, sd = 1)

  dat_full <- data.frame(X1 = X1, X2 = X2, Y = Y)

  # 2. Introduce missingness in X2 based on X1
  prob_missing <- plogis(X2 + rnorm(n,1,1),
                         location = 0, scale = 1)  # Logistic function for missingness)
  is_missing <- rbinom(n, size = 1, prob = prob_missing)

  dat_missing <- dat_full
  dat_missing$X1[is_missing == 1] <- NA

  ## Oracle: regression on full data (no missingness)
  fit_oracle <- lm(Y ~ X1 + X2, data = dat_full)
  coef_oracle <- coef(summary(fit_oracle))[-1, ]  # drop intercept
  results_oracle[sim, ] <- c(coef_oracle[, "Estimate"], coef_oracle[, "Std. Error"])

  ## Complete case: regression on observed data only
  fit_complete <- lm(Y ~ X1 + X2, data = dat_missing, na.action = na.omit)
  coef_complete <- coef(summary(fit_complete))[-1, ]
  results_complete[sim, ] <- c(coef_complete[, "Estimate"], coef_complete[, "Std. Error"])

  ## Multiple Imputation
  imp <- mice(dat_missing[,c("X1","X2")], m = 5, printFlag = FALSE)
  fit_mi <- with(imp, lm(Y ~ X1 + X2))
  pooled <- pool(fit_mi)
  summary_pool <- summary(pooled)

  results_mi[sim, ] <- c(summary_pool$estimate[2:3], summary_pool$std.error[2:3])
}

# Convert to dataframes
results_oracle <- as.data.frame(results_oracle)
results_complete <- as.data.frame(results_complete)
results_mi <- as.data.frame(results_mi)


############### check coverage
# Function to calculate empirical coverage over a range of CI levels
calculate_coverage_curve <- function(results, true_value, varname) {

  # Extract estimate and standard error
```

```r
  beta_hat <- results[[paste0("beta_", varname)]]
  se_hat <- results[[paste0("se_", varname)]]

  nominal_levels <- 1:99  # 1% to 99%
  coverage_rates <- numeric(length(nominal_levels))

  for (i in seq_along(nominal_levels)) {
    level <- nominal_levels[i] / 100  # convert to proportion
    t_quantile <- qt(1 - (1 - level) / 2, df = n - 3)  # t-distribution quantile

    lower_bound <- beta_hat - t_quantile * se_hat
    upper_bound <- beta_hat + t_quantile * se_hat


    # Check if true value lies inside the interval
    covered <- (lower_bound <= true_value) & (upper_bound >= true_value)
    coverage_rates[i] <- mean(covered) * 100  # percentage
  }

  data.frame(
    Nominal_Coverage = nominal_levels,
    Empirical_Coverage = coverage_rates
  )
}

## for X1
true_beta_X1 <- 2
oracle_curve_X1 <- calculate_coverage_curve(results_oracle, true_beta_X1, "X1")
complete_curve_X1 <- calculate_coverage_curve(results_complete, true_beta_X1, "X1")
mi_curve_X1 <- calculate_coverage_curve(results_mi, true_beta_X1, "X1")

## plot
# Plot coverage curve
pdf("./Downloads/x1_coverage.pdf", width = 6, height = 4)
plot(oracle_curve_X1$Nominal_Coverage,
     oracle_curve_X1$Empirical_Coverage, type = "l",
     col = "black", lwd = 2,
     ylim = c(0, 100),
     xlab = "Nominal Confidence Level (%)", ylab = "Empirical Coverage (%)",
     main = "Coverage Curve for X1 ")
lines(complete_curve_X1$Nominal_Coverage,
      complete_curve_X1$Empirical_Coverage, col = "#4a1564", lwd = 2)
lines(mi_curve_X1$Nominal_Coverage,
      mi_curve_X1$Empirical_Coverage, col = "#A85E83", lwd = 2)
legend("topleft", legend = c("oracle", "complete","MI"),
       col = c("black", "#4a1564","#A85E83"),
       lwd = 2, bty = "n")
#abline(0, 1, lty = 2, col = "gray")  # Ideal 45-degree line
dev.off()
```

```
## pdf
##   2
```

```r
###################
## for X1
true_beta_X2 <- -1
oracle_curve_X2 <- calculate_coverage_curve(results_oracle, true_beta_X2, "X2")
complete_curve_X2 <- calculate_coverage_curve(results_complete, true_beta_X2, "X2")
mi_curve_X2 <- calculate_coverage_curve(results_mi, true_beta_X2, "X2")

pdf("./Downloads/x2_coverage.pdf", width = 6, height = 4)
plot(oracle_curve_X2$Nominal_Coverage,
     oracle_curve_X2$Empirical_Coverage, type = "l",
     col = "black", lwd = 2,
     ylim = c(0, 100),
     xlab = "Nominal Confidence Level (%)", ylab = "Empirical Coverage (%)",
     main = "Coverage Curve for X2 ")
lines(complete_curve_X2$Nominal_Coverage,
      complete_curve_X2$Empirical_Coverage, col = "#4a1564", lwd = 2)
lines(mi_curve_X2$Nominal_Coverage,
      mi_curve_X2$Empirical_Coverage, col = "#A85E83", lwd = 2)
legend("topleft", legend = c("oracle", "complete","MI"),
       col = c("black", "#4a1564","#A85E83"),
       lwd = 2, bty = "n")
#abline(0, 1, lty = 2, col = "gray")  # Ideal 45-degree line
dev.off()
```

```
## pdf
##   2
```

```r
################################
pdf("./Downloads/x1_density.pdf", width = 6, height = 4)
plot(density(results_oracle$beta_X1),
     col = "black", lwd = 2,
     xlim = c(1, 2.5),
     xlab = "Estimate", ylab = "Density",
     main = "Distribution of Estimates for X1")
lines(density(results_mi$beta_X1), col = "#A85E83", lwd = 2)
legend("topright", legend = c("oracle", "MI"),
       col = c("black", "#A85E83"),
       lwd = 2, bty = "n")
abline(v=true_beta_X1, lty = 2,lwd=2, col = "gray")  # True value line
dev.off()
```

```
## pdf
##   2
```

```r
pdf("./Downloads/x2_density.pdf", width = 6, height = 4)
plot(density(results_oracle$beta_X2),
     col = "black", lwd = 2,
     xlim = c(-1.2, -0.7),
     xlab = "Estimate", ylab = "Density",
     main = "Distribution of Estimates for X2")
lines(density(results_mi$beta_X2), col = "#A85E83", lwd = 2)
legend("topright", legend = c("oracle", "MI"),
```

```
       col = c("black", "#A85E83"),
       lwd = 2, bty = "n")
abline(v=true_beta_X2, lty = 2,lwd=2, col = "gray")  # True value line
dev.off()
```

```
## pdf
##   2
```

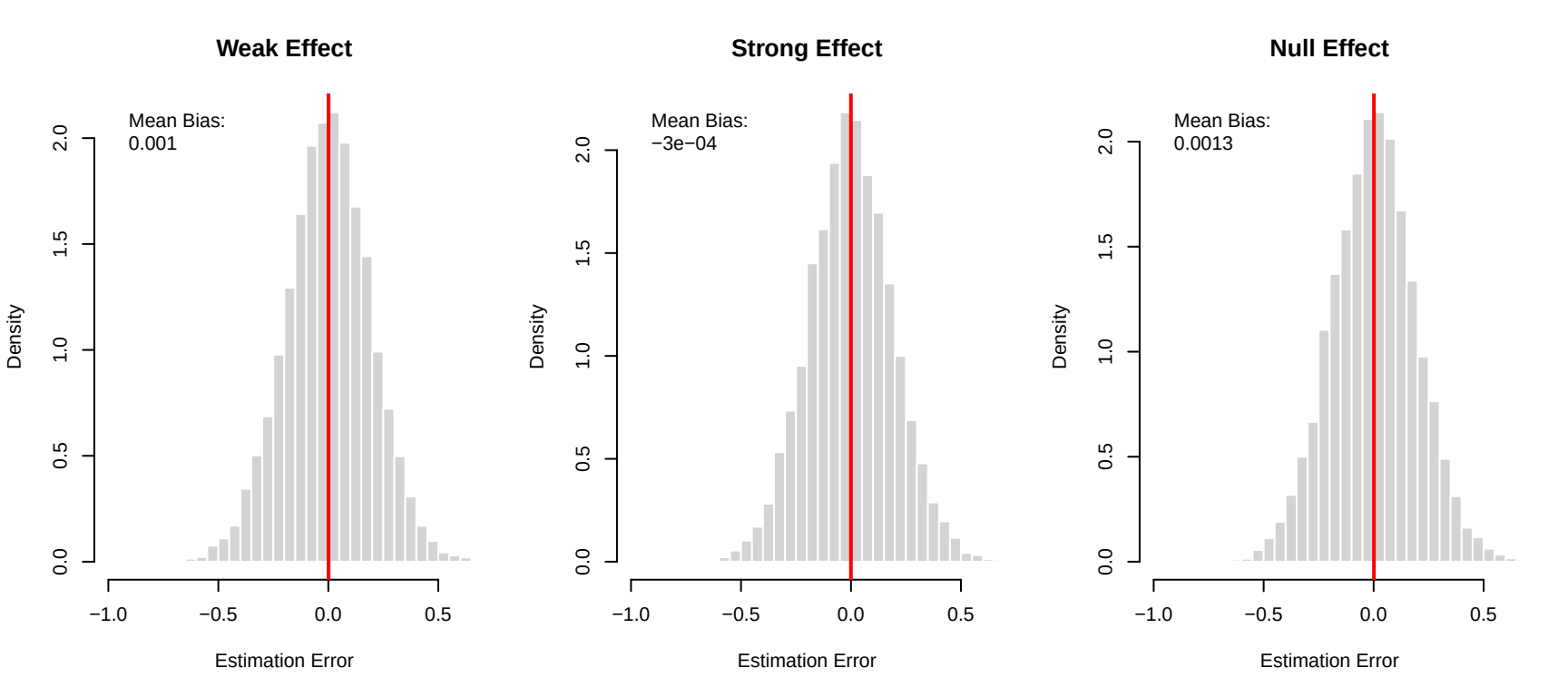
d. The MI estimator is very biased and fails to cover. The other two estimators are unbiased. Notice that if you include Y in the MI model, the MI estimator would also seem to work well. However, this does not make sense during research, especially if you are making a causal claim. You cannot use Y to predict the missing values of X and then use the predicted X to estimate the effects of X on Y, which would cause circular dependence. But for this exam, we regard either way correct.

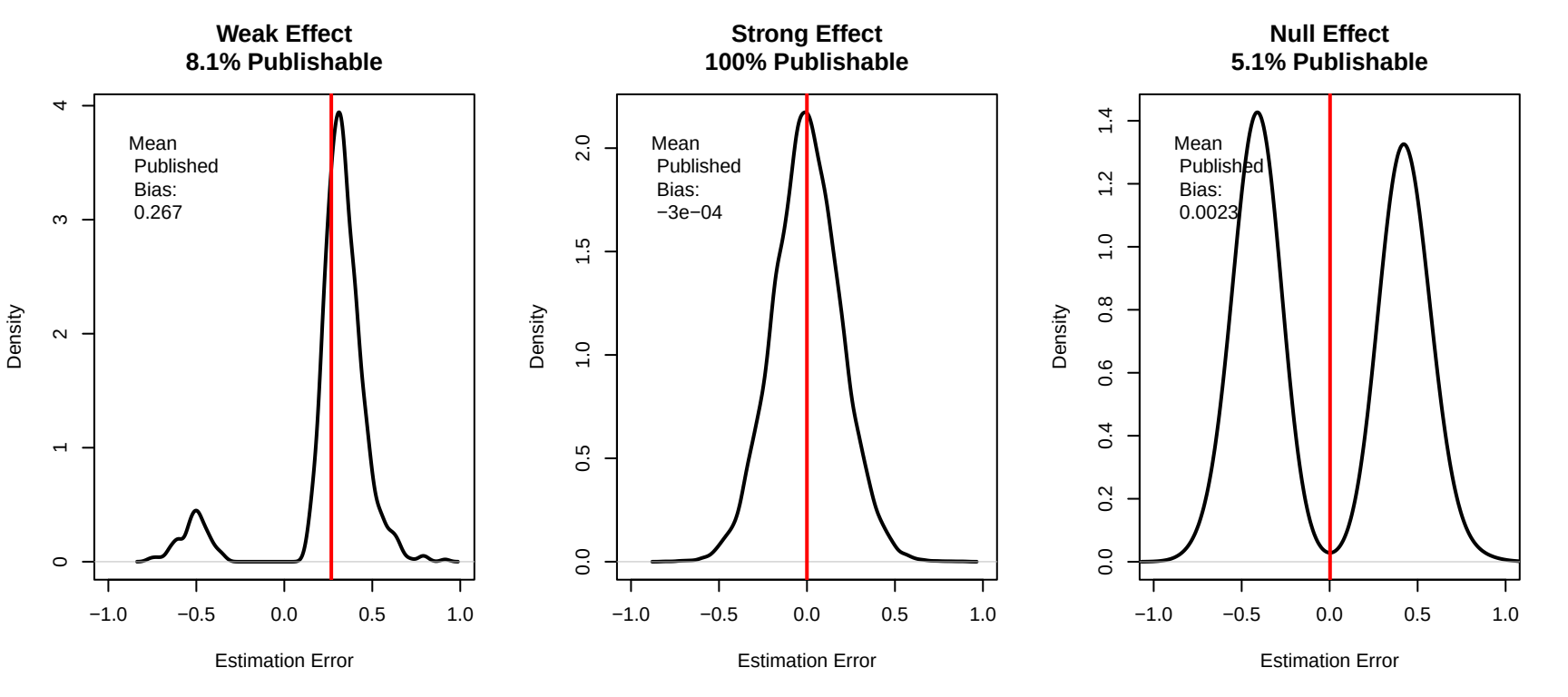
e. The most realistic and reliable estimator here is the complete case estimator, since MI is biased and we cannot observe the missing values in reality. The reason is that under bivariate OLS with one of the variables missing at random, the missing and observed groups are very similar. We can check the distribution of X_2 and Y by X_1 missing or observed in a simulated data frame (you can also check the mean and sd difference over simulations):

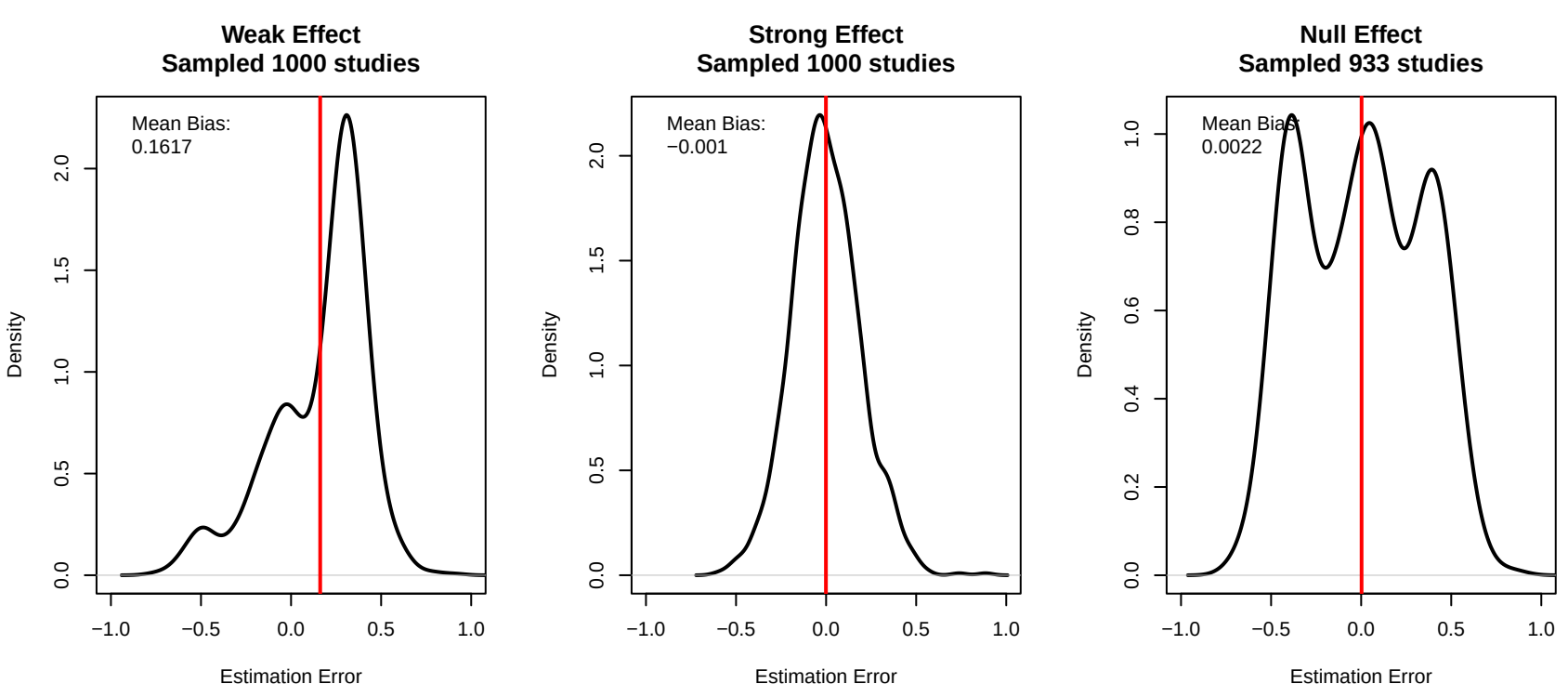
```
dat_missing <- dat_missing %>%
  mutate(x1_missing = ifelse(is.na(X1), "Missing", "Observed"))

ggplot(dat_missing, aes(x = X2, fill = x1_missing)) +
  geom_density(alpha = 0.5) +
  labs(title = "Distribution of x2 by x1 Missingness",
       x = "x2",
       fill = "x1 Status") +
  theme_minimal()
```

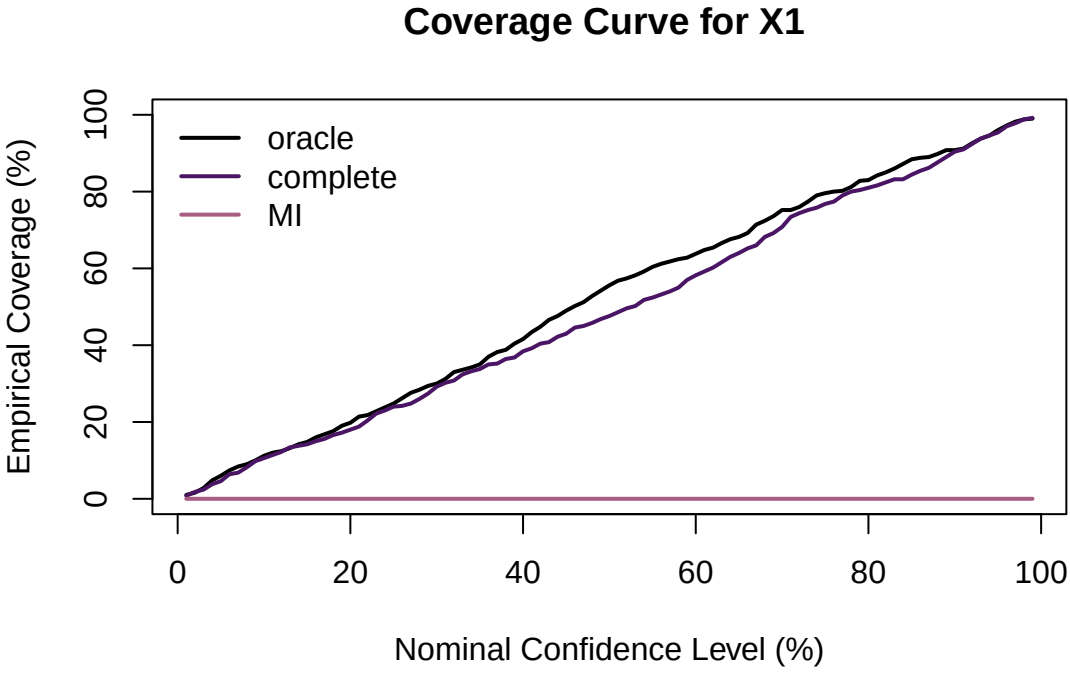

```
ggplot(dat_missing, aes(x = Y, fill = x1_missing)) +
  geom_density(alpha = 0.5) +
  labs(title = "Distribution of y by x1 Missingness",
       x = "y",
       fill = "x1 Status") +
  theme_minimal()
```

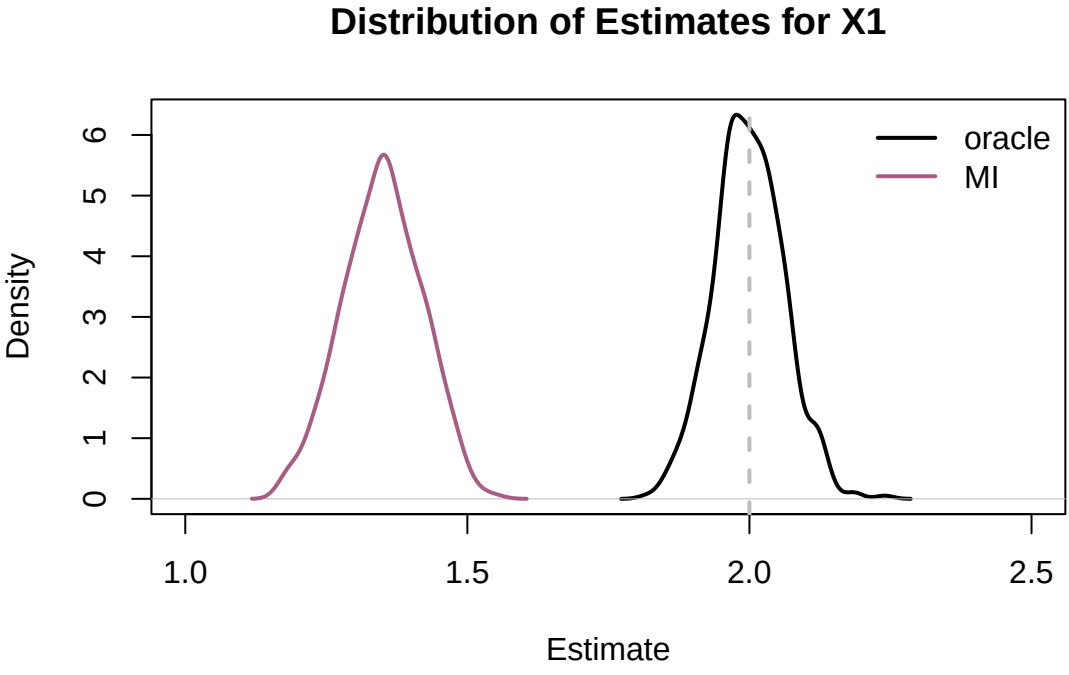

Distribution of y by x1 Missingness
density
0.2
0.1
0.0
−10.0
−7.5
−5.0
−2.5
0.0
y
x1 Status
Missing
Observed

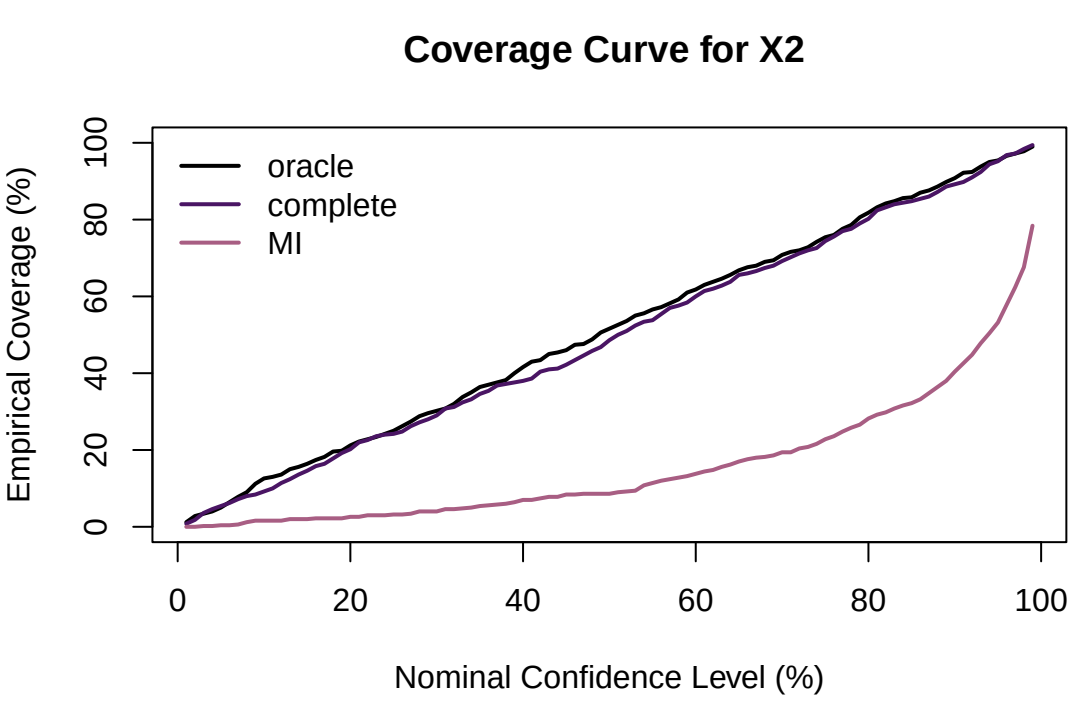

Weak Effect
Mean Bias:
0.001
Density
0.0
0.5
1.0
1.5
2.0
−1.0
−0.5
0.0
0.5
Estimation Error
Strong Effect
Mean Bias:
−3e−04
Density
0.0
0.5
1.0
1.5
2.0
−1.0
−0.5
0.0
0.5
Estimation Error
Null Effect
Mean Bias:
0.0013
Density
0.0
0.5
1.0
1.5
2.0
−1.0
−0.5
0.0
0.5
Estimation Error

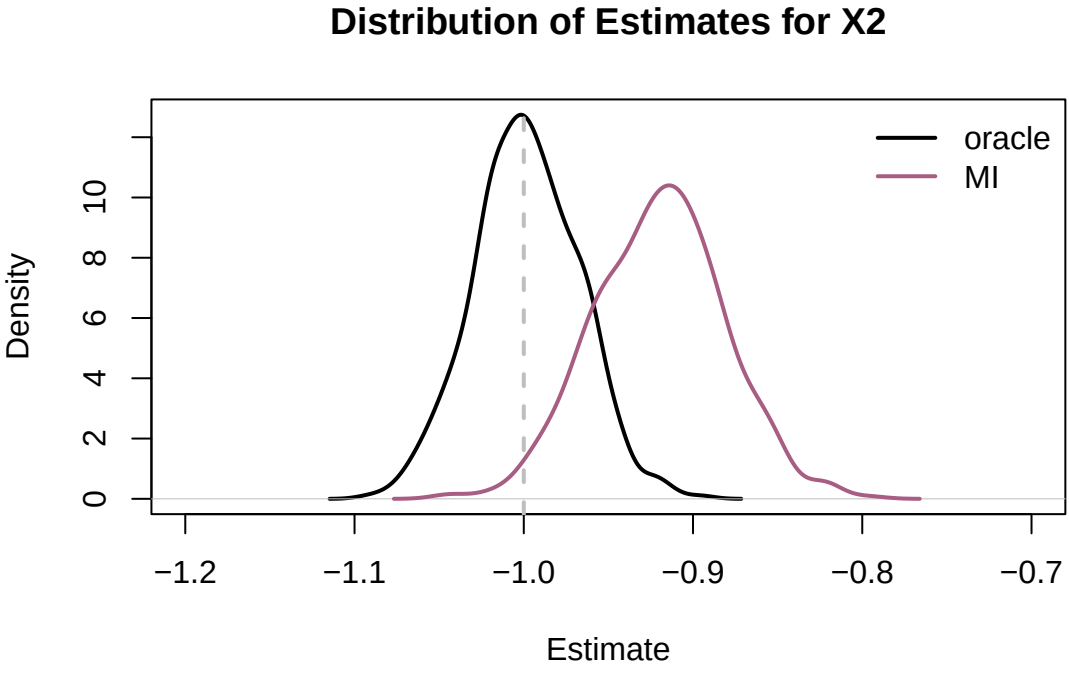

Weak Effect
8.1% Publishable
Mean
Published
Bias:
0.267
Density
Estimation Error
Strong Effect
100% Publishable
Mean
Published
Bias:
−3e−04
Density
Estimation Error
Null Effect
5.1% Publishable
Mean
Published
Bias:
0.0023
Density
Estimation Error


Weak Effect
Sampled 1000 studies
Mean Bias:
0.1617
Density
Estimation Error
Strong Effect
Sampled 1000 studies
Mean Bias:
−0.001
Density
Estimation Error
Null Effect
Sampled 933 studies
Mean Bias:
0.0022
Density
Estimation Error


Coverage Curve for X1
oracle
complete
MI
Empirical Coverage (%)
0
20
40
60
80
100
Nominal Confidence Level (%)


Distribution of Estimates for X1
Density
0
1
2
3
4
5
6
1.0
1.5
2.0
2.5
Estimate
oracle
MI


Coverage Curve for X2
oracle
complete
MI
Empirical Coverage (%)
0
20
40
60
80
100
Nominal Confidence Level (%)


Distribution of Estimates for X2
oracle
MI
Density
0
2
4
6
8
10
−1.2
−1.1
−1.0
−0.9
−0.8
−0.7
Estimate