STA 6384, Report 3.4

Carson Slater *Baylor University*

Problem:

1.

For sample sizes $n = 20, 50$, and 100, generate 1000 tables of the form in Table 3.2. Do this for $\theta = 0.1, 0.5$, and 0.75. The 9 combination vectors, $\delta = (n, \theta)$, constitute your design points for this simulation study. Each generated table is called a *replication* at the design point δ. So, you will have 1000 replications at each of the 9 values of δ.

Based on the provided context, we are dealing with an odds ratio θ derived from a 2x2 contingency table of the form:

	Group 1	Group 2
Outcome 1	n_{11}	n_{12}
Outcome 2	n_{21}	n_{22}

The odds ratio is defined as $\theta = \frac{n_{11}n_{22}}{n_{12}n_{21}}$. To simulate these tables, we will make the following assumptions:

1. **Study Design**: We assume a design with two groups of fixed sizes, n_1 and n_2. The total sample size n is split evenly between the groups, so $n_1 = n_2 = n/2$. This is a reasonable assumption given the choice of even sample sizes (20, 50, 100).

2. **Data Generation**: The counts for an outcome of interest in each group, n_{11} and n_{21}, are simulated from two independent binomial distributions:

$$n_{11} \sim \text{Binomial}(n_1, p_1)$$
$$n_{21} \sim \text{Binomial}(n_2, p_2)$$

3. **Probabilities**: The event probabilities p_1 and p_2 must satisfy the given odds ratio θ. The relationship is:

$$\theta = \frac{p_1/(1-p_1)}{p_2/(1-p_2)}$$

   As there are infinite pairs (p_1, p_2) for a given θ, we fix a baseline probability for the second group at $p_2 = 0.20$ and solve for p_1. The resulting formula for p_1 is:

$$p_1 = \frac{\theta \cdot p_2}{1 - p_2 + \theta \cdot p_2}$$

This framework allows us to generate 1,000 tables for each of the 9 design points $\delta = (n, \theta)$. We note that this procedure may generate cells with zero counts, an issue that must be handled in subsequent calculations of the sample odds ratio.

We generate data for 9 design points, which are the combinations of sample sizes $n \in \{20, 50, 100\}$ and true odds ratios $\theta \in \{0.1, 0.5, 0.75\}$. For each design point, we generate 1,000 replications, where each replication is a 2x2 contingency table.

The code below defines the simulation parameters and then generates the full dataset. We assume the total sample size n is split into two equal groups ($n_1 = n_2 = n/2$). We fix a baseline event probability $p_2 = 0.2$ and calculate the required probability p_1 to achieve the target θ.

```
## Simulation complete. Total rows: 9000
```

```
## The first few rows of the generated data are:
```

```
## # A tibble: 6 x 7
##       n theta replication_id   n11   n12   n21   n22
##   <dbl> <dbl>          <int> <int> <dbl> <int> <dbl>
## 1    20   0.1              1     0    10     4     6
## 2    20   0.1              2     0    10     3     7
## 3    20   0.1              3     1     9     2     8
## 4    20   0.1              4     0    10     2     8
## 5    20   0.1              5     0    10     1     9
## 6    20   0.1              6     0    10     4     6
```

```
##
##
## The last few rows of the generated data (for n=100, theta=0.75) are:
```

```
## # A tibble: 6 x 7
##       n theta replication_id   n11   n12   n21   n22
##   <dbl> <dbl>          <int> <int> <dbl> <int> <dbl>
## 1   100  0.75            995     7    43    11    39
## 2   100  0.75            996    10    40    12    38
## 3   100  0.75            997     9    41     8    42
## 4   100  0.75            998     6    44    11    39
## 5   100  0.75            999     5    45     9    41
## 6   100  0.75           1000     8    42    12    38
```

2.

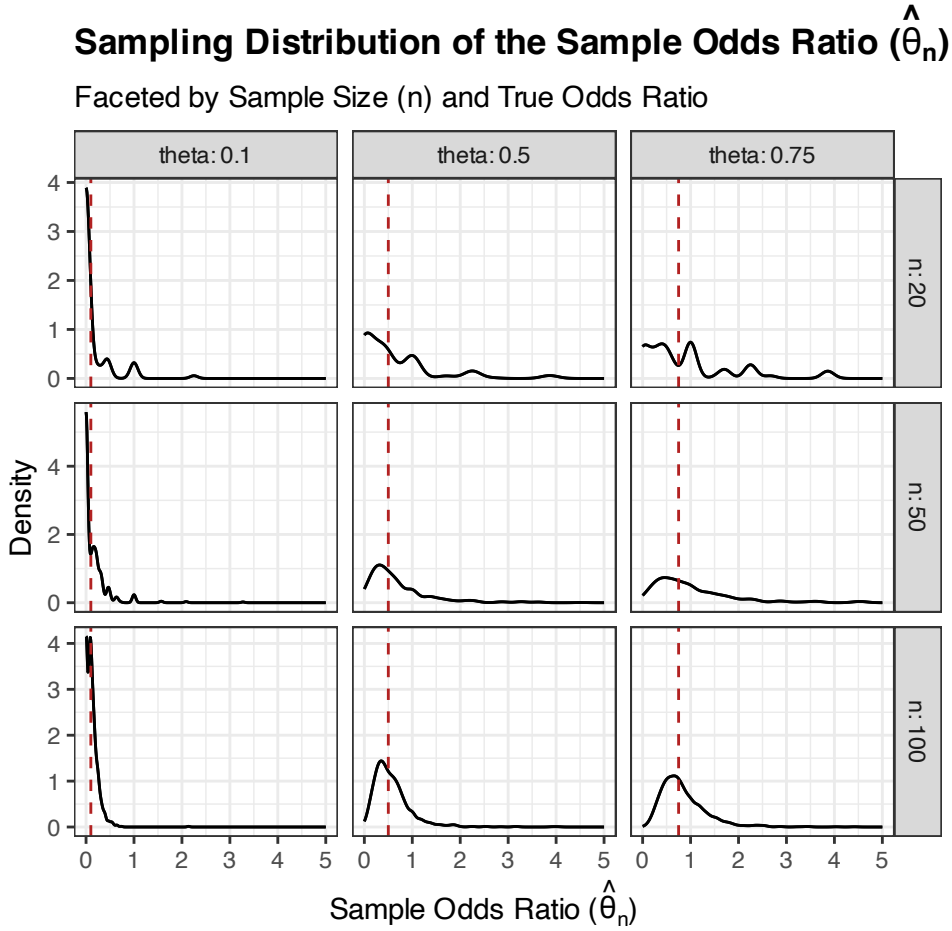
At each design point, δ, and for each replication, compute the following: (a) $\hat{\theta}_n$, (b) $\log \hat{\theta}_n$, (c) $\hat{\sigma}(\log \hat{\theta}_n)$, (d) the exponentiated Wald CI bounds using (3.1.3) for each table. (e) whether or not the "true" value of θ is in the Wald interval.

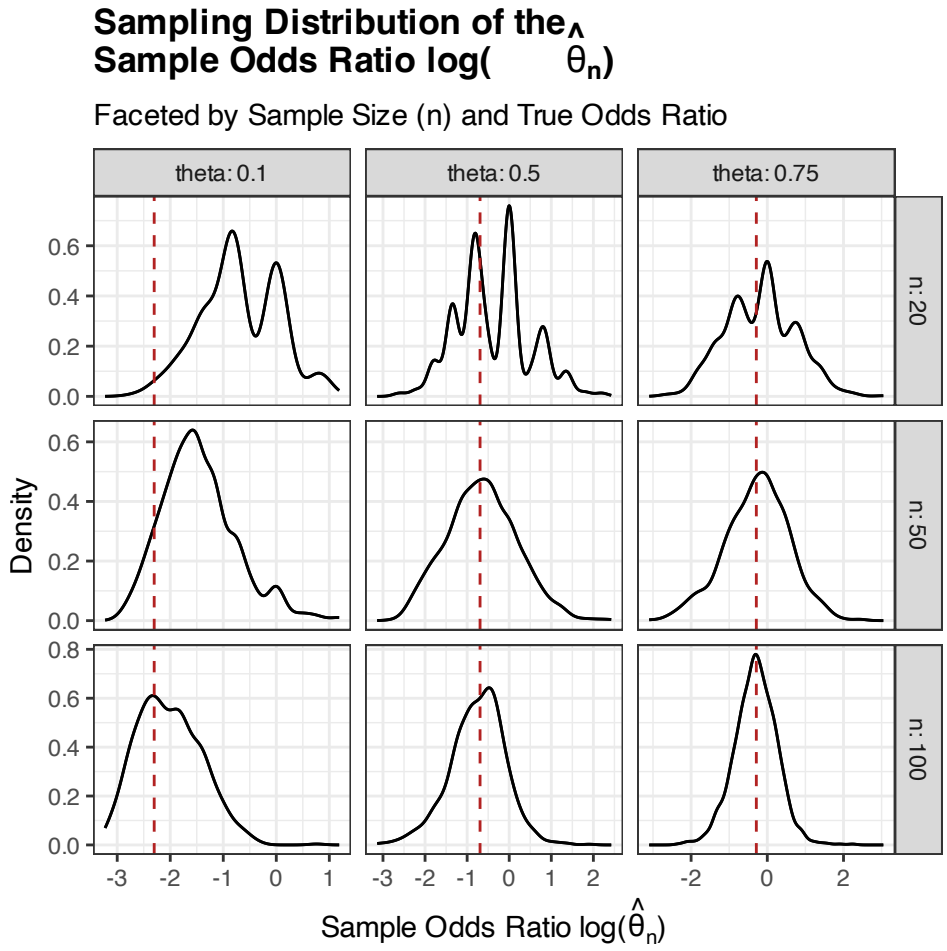
Table 1: Coverage of 95% Wald CI (No Continuity Correction)

Sample Size (n)	True Theta	Coverage Probability
20	0.10	0.911
20	0.50	0.985
20	0.75	0.992
50	0.10	0.935
50	0.50	0.962
50	0.75	0.974
100	0.10	0.954
100	0.50	0.961
100	0.75	0.959

That will give you 9 sets of 1000 values of $\hat{\theta}_n, \log \hat{\theta}_n$, etc. The number of times your intervals contain the "true" value divided by 1000 will be your *coverage* for that design point.

3. For each δ in your simulation, exhibit the following: (a) kernel smoothed histogram of the sampling distribution of $\hat{\theta}_n$, (b) kernel smoothed histogram of the sampling distribution of $\log \hat{\theta}_n$,





Code Appendix

```
knitr::opts_chunk$set(dev = "cairo_pdf",
                      fig.width = 5,
                      fig.height = 5,
                      fig.align = 'center',
                      echo = FALSE,
                      message = FALSE,
                      warning = TRUE,
                      error = FALSE)
library("tidyverse"); library("patchwork"); library("glue")
library("scales", warn.conflicts = FALSE); library("extrafont")
library("tinytex"); library("patchwork"); library("knitr")
library("tidyr"); library("latex2exp")
# library("furrr"); library("future")

theme_set(theme_minimal(base_family = "Roboto Condensed"))
```

```r
conflicted::conflicts_prefer(
  readr::col_factor(),
  purrr::discard(),
  rstan::extract(),
  dplyr::lag(),
  rstan::traceplot(),
  viridis::viridis_pal(),
  readr::parse_date(),
  kableExtra::group_rows(),
  gridExtra::combine(),
  rstan::extract
)
# 1. Define Simulation Parameters
# ---------------------------------
n_values <- c(20, 50, 100)
theta_values <- c(0.1, 0.5, 0.75)
n_reps <- 1000
p2_baseline <- 0.20 # Fixed baseline probability for the second group

# Create a data frame of all 9 design points
design_points <- expand.grid(n = n_values, theta = theta_values)

# 2. Generate Data for All Replications
# ---------------------------------------

# Use a loop or map function to iterate through each design point
# We will create a list of data frames first, and then combine them.
all_reps_list <- apply(design_points, 1, function(row) {

  # Extract parameters for the current design point
  n_total <- as.numeric(row['n'])
  theta_true <- as.numeric(row['theta'])

  # Calculate required p1 based on the true theta and baseline p2
  p1_true <- (theta_true * p2_baseline) / (1 - p2_baseline + theta_true * p2_

  # Define group sizes
  n1 <- n_total / 2
  n2 <- n_total / 2

  # Run 1000 replications for this design point
  replications <- tibble(
    replication_id = 1:n_reps,
    n11 = rbinom(n_reps, size = n1, prob = p1_true),
    n21 = rbinom(n_reps, size = n2, prob = p2_baseline)
  ) |>
  mutate(
```

```
    # Calculate the other two cells of the 2x2 table
    n12 = n1 - n11,
    n22 = n2 - n21
  )

  replications
})

# Combine the list of data frames into one, adding columns for n and theta
simulation_data <- bind_rows(all_reps_list, .id = "design_point_id") |>
  # Join with original design points to get n and theta columns
  left_join(mutate(design_points, design_point_id = as.character(row_number()
  select(n, theta, replication_id, n11, n12, n21, n22)

# 3. Display the Structure of the Final Dataset
# ---------------------------------------------
cat("Simulation complete. Total rows:", nrow(simulation_data), "\n")
cat("The first few rows of the generated data are:\n\n")
head(simulation_data)

cat("\n\nThe last few rows of the generated data (for n=100, theta=0.75) are:
tail(simulation_data)
# --- Perform Calculations for Part 2 (No Correction) ---
z_alpha <- qnorm(0.975)

# Calculate statistics for each replication and then summarize coverage
# NOTE: The 'mutate' step for correction has been removed.
sim_data <- simulation_data |>
  # Compute statistics directly on raw counts
  mutate(
    theta_hat = (n11 * n22) / (n12 * n21),
    log_theta_hat = log(theta_hat),
    se_log_theta_hat = sqrt(1/n11 + 1/n12 + 1/n21 + 1/n22),
    ci_lower = exp(log_theta_hat - z_alpha * se_log_theta_hat),
    ci_upper = exp(log_theta_hat + z_alpha * se_log_theta_hat),
    is_covered = (theta >= ci_lower & theta <= ci_upper)
  )
  # Group by design point to calculate the final coverage probability
coverage_results_no_corr <- sim_data |>
  group_by(n, theta) |>
  summarise(
    coverage = mean(is_covered, na.rm = TRUE),
    .groups = 'drop'
  )

# --- Present Results in a Kable ---
kable(coverage_results_no_corr,
```

```
      caption = "Coverage of 95% Wald CI (No Continuity Correction)",
      col.names = c("Sample Size (n)", "True Theta", "Coverage Probability"),
      digits = 3,
      align = 'c')
sim_data |>
  ggplot(aes(x = theta_hat)) +
  geom_density(aes(y = after_stat(density)), bins = 40, color = "black", alph
  geom_density() +
  geom_vline(aes(xintercept = theta), color = "firebrick", linetype = "dashed
  xlim(c(0, 5)) +
  facet_grid(n ~ theta, labeller = label_both, scales = "free") +
  labs(
    title = expression(bold("Sampling Distribution of the Sample Odds Ratio (
    subtitle = "Faceted by Sample Size (n) and True Odds Ratio",
    x = expression("Sample Odds Ratio (" * hat(theta)[n] * ")"),
    y = "Density"
  ) +
  theme_bw()

sim_data |>
  ggplot(aes(x = log_theta_hat)) +
  geom_density(aes(y = after_stat(density)), bins = 40, color = "black", alph
  geom_density() +
  geom_vline(aes(xintercept = log(theta)), color = "firebrick", linetype = "da
  facet_grid(n ~ theta, labeller = label_both, scales = "free") +
  labs(
    title = expression(bold("Sampling Distribution of the\nSample Odds Ratio
    subtitle = "Faceted by Sample Size (n) and True Odds Ratio",
    x = expression("Sample Odds Ratio log(" * hat(theta)[n] * ")"),
    y = "Density"
  ) +
  theme_bw()
```