Likelihood ratio interval:
find the μ such that
$2[\ell(\hat{\mu}) - \ell(\mu)] = (1.96)^2$.
the interval formed by the 2 μs (solution) gives the 95% CI for μ.

$$f(x_i; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right)$$

The MLE for ( ) minimizes (()):

$$\hat{\mu} = \arg\min_{\mu} \ell(\mu)$$

The likelihood interval is calculated as:

$$\text{Likelihood Interval (Likelihood Ratio)} = [\hat{\mu} - 1.96 \times SE, \hat{\mu} + 1.96 \times SE]$$ X

4) Non-Parametric Bootstrap Interval

The non-parametric bootstrap interval is calculated as the 2.5th and 97.5th percentiles of the bootstrap sample means:

$$\text{Non-Parametric Bootstrap Interval} = \text{Quantile}(\text{means}, 0.025, 0.975)$$

Functions for Estimators

```r
# 1) T-Interval
calculate_t_interval <- function(means) {
  quantile(means, c(0.025, 0.975))
}

# 2) Likelihood Interval Based on MLE
calculate_likelihood_interval_mle <- function(data) {
  mle <- mean(data)
  n <- length(data)
  se <- sd(data) / sqrt(n)
  c(mle - 1.96 * se, mle + 1.96 * se)
}
```

X

```r
# 3) Likelihood Interval Based on Likelihood Ratio
calculate_likelihood_interval_lr <- function(data) {
  log_likelihood <- function(mean) {
    -sum(dnorm(data, mean = mean, sd = sd(data), log =
  }
  mle_mean <- optimize(log_likelihood, interval = range
  n <- length(data)
  se <- sd(data) / sqrt(n)
  c(mle_mean - 1.96 * se, mle_mean + 1.96 * se)
}
```



```r
# 4) Non-Parametric Bootstrap Interval
calculate_nonparametric_interval <- function(means) {
  quantile(means, c(0.025, 0.975))
}
```

Expected Analytic Values

```r
# Define expected analytic values for each distribution
expected_analytic_values <- function(dist_name, n) {
  switch(dist_name,
         "normal" = list(mean = 0, se = 1 / sqrt(n)),
         "t" = list(mean = 0, se = sqrt(10 / (10 - 2))
         "gamma" = list(mean = 2 / 1, se = sqrt(2 / (1^
         "lognormal" = list(mean = exp(0 + (1^2) / 2),
         "weibull" = list(mean = 2 * gamma(1 + 1 / 2),
  )
}

# Generate a table of analytic values
generate_analytic_table <- function(m_list, n_list, dis
  analytic_table <- do.call(rbind, lapply(n_list, funct
    do.call(rbind, lapply(distributions, function(dist_
      analytic_values <- expected_analytic_values(dist_
      analytic_mean <- analytic_values$mean
      analytic_se <- analytic_values$se
      analytic_t_interval <- c(analytic_mean - 1.96 * a

      data.frame(
        n = n_value,
        distribution = dist_name,
        analytic_mean = analytic_mean,
        analytic_se = analytic_se,
        analytic_t_interval_lower = analytic_t_interval
        analytic_t_interval_upper = analytic_t_interval
      )
    }))
  }))
  return(analytic_table)
}

analytic_table <- generate_analytic_table(m_list, n_lis

# Display the analytic table
```

```
print(head(analytic_table,5))
```

Have problem reading results. Can they be lined up properly? OR in spreadsheet?

```
  n distribution analytic_mean analytic_se
analytic_t_interval_lower
1 10       normal      0.000000   0.3162278
-0.6198064
2 10            t      0.000000   0.3535534
-0.6929646
3 10        gamma      2.000000   0.4472136
1.1234614
4 10    lognormal      1.648721   0.6834306
0.3091972
5 10      weibull      1.772454   0.2929859
1.1982015
  analytic_t_interval_upper
1                 0.6198064
2                 0.6929646
3                 2.8765386
4                 2.9882453
5                 2.3467062
```

Bootstrap Function

```
# Usage within the bootstrap_intervals function
bootstrap_intervals <- function(data, m) {
  n <- length(data)
  means <- numeric(m)
  for (i in 1:m) {
    sample_data <- sample(data, size = n, replace = TRU
    means[i] <- mean(sample_data)
  }

  list(
    t_interval = calculate_t_interval(means),
    likelihood_interval_mle = calculate_likelihood_inte
    likelihood_interval_lr = calculate_likelihood_inter
    nonparametric_interval = calculate_nonparametric_in
  )


}
```


Iterate through all combinations of each distribution

```
results <- list()

for (m in m_list) {
  for (n in n_list) {
    for (dist_name in distributions) {
      set.seed(123)  # Ensure reproducibility
      data <- simulate_distributions(dist_name, n)
      intervals <- bootstrap_intervals(data, m)
      results[[paste("m", m, "n", n, dist_name, sep = "

    }
  }
}
```

Bootstrap Results

```
# Define expected parameter values for each distribu  .o
distribution_params <- list(
  normal = list(mean = 0, sd = 1),
  t = list(df = 10),
  gamma = list(shape = 2, rate = 1),
  lognormal = list(meanlog = 0, sdlog = 1),
  weibull = list(shape = 2, scale = 1)
)

# Convert results list to a data frame with parameter v
results_table <- do.call(rbind, lapply(names(results),
  # Extract m, n, and distribution from the name
  split_name <- strsplit(name, "_")[[1]]
  m_value <- as.numeric(split_name[2])
  n_value <- as.numeric(split_name[4])
  dist_name <- split_name[5]

  # Extract intervals
  intervals <- results[[name]]

  # Extract parameters for the distribution
  params <- distribution_params[[dist_name]]

  BS_mean <- mean(intervals$t_interval)
```