ST502 Project 1

Group E: Saurabh Gupta, Franklin Zhou

2024-10-04

Introduction

Our goal of this project is to compare different confidence interval procedures that attempt to capture the p, the true probability in a binomial distribution.

Methods

In this project, we compare the performance of six methods with different combination of sample size n and probability p. These six methods are:

- Wald interval
- Adjusted Wald interval
- Clopper-Pearson (exact) interval
- Score interval
- Raw percentile interval using a parametric bootstrap
- Bootstrap t interval using a parametric bootstrap

Wald interval function

The formula of calculating Wald interval is:

$$\hat{p} - z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} < p < \hat{p} + z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}},$$

where the z_c denotes the c quantile of the standard normal distribution.

To calculate the Wald interval for p, we need to use sample's probability $\hat{p}$, sample size n and the confidence level $1-\alpha$. The function of calculating Wald Interval is given below:

```
WaldCI <- function(y, n, alpha = 0.05){
  p_hat = y/n  # Estimate of the probability of success
  # Return lower and upper bound of Wald CI
  c("wald ci lower" = p_hat - qnorm(1-alpha/2)*sqrt((p_hat*(1-p_hat))/n),
    "wald ci upper" = p_hat + qnorm(1-alpha/2)*sqrt((p_hat*(1-p_hat))/n))
}
```

In this project, we use $\alpha = 0.05$ as default in all calculation.

Adjusted Wald interval function

To overcome the poor performance of Wald interval, adjusted Wald interval was proposed. The formula of adjusted Wald interval is almost the same as Wald interval, except that we add 2 successes and 2 failures to the sample data set:

$$\tilde{p} - z_{1-\frac{\alpha}{2}}\sqrt{\frac{\tilde{p}(1-\tilde{p})}{n}} < p < \tilde{p} + z_{1-\frac{\alpha}{2}}\sqrt{\frac{\tilde{p}(1-\tilde{p})}{n}},$$

where $\tilde{p} = (Y+2)/(n+4)$.

The function of calculating adjusted Wald Interval is given below:

```
AdjWaldCI <- function(y, n, alpha = 0.05){
  p_tilde = (y+2)/(n+4)  # Adjusted estimate of probability of success
  # Return lower and upper bound of Adjusted Wald CI
  c("adj wald ci lower" = p_tilde - qnorm(1-alpha/2)*sqrt((p_tilde*(1-p_tilde))/n),
    "adj wald ci upper" = p_tilde + qnorm(1-alpha/2)*sqrt((p_tilde*(1-p_tilde))/n))
}
```

Clopper-Pearson (exact) interval function

Clopper-Pearson interval is referred to as "exact" interval. This interval is guaranteed to have coverage probability of at least $1-\alpha$ for every value of p. When $y = 1, 2, \cdots, n-1$, the confidence interval is:

$$\left[1 + \frac{n-y+1}{yF_{2y,2(n-y+1),\frac{\alpha}{2}}}\right]^{-1} < p < \left[1 + \frac{n-y}{(y+1)F_{2(y+1),2(n-y),1-\frac{\alpha}{2}}}\right]^{-1},$$

where the $F_{a,b,c}$ denotes the c quantile from the F distribution with degrees of freedom a and b.

When $y = 0$, we set the interval to be $[0, 0]$ and When $y = n$, we set the interval to be $[1, 1]$. The function of calculating Clopper-Pearson interval is given below:

```
ClopperPearsonCI <- function(y, n, alpha = 0.05){
  # Handle special cases where y is 0 or y equals n
  if (y==0){
    c("C-P interval lower" = 0,
      "C-P interval upper" = 0)
    } else if (y==n){
      c("C-P interval lower" = 1,
        "C-P interval upper" = 1)
    } else {
      # General case for Clopper-Pearson CI using F-distribution
  c("C-P interval lower" = 1 / (1 + ((n-y+1) / (y * qf(alpha/2, 2*y, 2*(n-y+1))))),
    "C-P interval upper" = 1 / (1 + ((n-y) / ((y+1) * qf(1-(alpha/2), 2*(y+1), 2*(n-y))))))
    }
}
```

Score interval function

Score interval can be used with almost all sample sizes and p values. The formula is given below:

$$\frac{\hat{p}+\frac{z^2_{1-\frac{\alpha}{2}}}{2n}-z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{p}(1-\hat{p})+\frac{z^2_{1-\frac{\alpha}{2}}}{4n}}{n}}}{1+\frac{z^2_{1-\frac{\alpha}{2}}}{n}}<p<\frac{\hat{p}+\frac{z^2_{1-\frac{\alpha}{2}}}{2n}+z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{p}(1-\hat{p})+\frac{z^2_{1-\frac{\alpha}{2}}}{4n}}{n}}}{1+\frac{z^2_{1-\frac{\alpha}{2}}}{n}},$$

where the z_c denotes the c quantile of the standard normal distribution.

And the function of calculating Score interval is given below:

```
ScoreCI <- function(y, n, alpha = 0.05){
  p_hat = y/n # Estimate of probability of success
  z = qnorm(1-alpha/2) # Z-score for the desired confidence level
  # Return lower and upper bound of Score CI
  c("Score interval lower" =
      (p_hat + z^2/(2*n) - z*sqrt((p_hat*(1-p_hat) + z^2/(4*n))/n)) / (1 + z^2/n),
    "Score interval upper" =
      (p_hat + z^2/(2*n) + z*sqrt((p_hat*(1-p_hat) + z^2/(4*n))/n)) / (1 + z^2/n))
}
```

Raw percentile interval and Bootstrap t interval function (both using a parametric bootstrap)

- To find the Raw percentile interval, we just need to find the $\frac{\alpha}{2}$ quantile and the $1-\frac{\alpha}{2}$ quantile of the $\hat{p}$ of samples.

$$\hat{p}_{\frac{\alpha}{2}}<p<\hat{p}_{1-\frac{\alpha}{2}}$$

where the $\hat{p}_c$ denotes the c quantile of the proportion of the samples.

- To find the Bootstrap t interval, we use following steps:

1. Calculate the mean of our sample's $\hat{p}$ to generate bootstrapped $\hat{p}_{boot}$,
2. Use the mean of bootstrapped $\hat{p}_{boot}$ to generate secondary bootstrapped $\hat{p}_{boot2}$,
3. Use secondary bootstrapped $\hat{p}_{boot2}$ to find the estimated standard error of $\hat{p}_{boot}$,
4. Calculate the interval based on formula.

$$\hat{p}-t_{1-\frac{\alpha}{2}}\cdot\hat{SE}<p<\hat{p}-t_{\frac{\alpha}{2}}\cdot\hat{SE}$$

where the t_c denotes the c quantile of the bootstrapped t values, and

$$T=\frac{\hat{p}-p}{\hat{SE}(\hat{p})}$$

The function of finding Raw percentile interval and Bootstrap t interval is given below:

```r
BootstrapCI <- function(x, n, size, B, alpha = 0.05){

   # Special case handling when x = 0 or x = size
  if (x == 0) {
    return(c(0, 0, 0, 0))  # Raw and t-bootstrap intervals set to [0, 0]
  } else if (x == size) {
    return(c(1, 1, 1, 1))  # Raw and t-bootstrap intervals set to [1, 1]
  } else {
    sample_p <- x/size # Estimate proportion

  # Calculate raw percentile interval from bootstrap samples
  raw_lower <- quantile(sample_p, alpha / 2, na.rm = TRUE)
  raw_upper <- quantile(sample_p, 1 - alpha / 2, na.rm = TRUE)
  raw_interval <- c(raw_lower, raw_upper)

  # Bootstrap t interval calculations
  p_hat <- mean(sample_p)
  boot_estimates <- replicate(B, {
    sim_data <- rbinom(n, size, prob = p_hat)  # Generate bootstrap sample
    p_hat_boot <- mean(sim_data / size)  # Bootstrap estimate of proportion

    # Perform secondary bootstrap for standard error estimation
    B2 <- 50
    secondary_boot <- replicate(B2, {
      sim_data2 <- rbinom(n, size, prob = p_hat_boot)  # Second bootstrap sample
      p_hat_boot2 <- mean(sim_data2 / size)  # Bootstrap estimate
      return(p_hat_boot2)
    })

    estimated_SE_p_hat_boot <- sd(secondary_boot)  # Standard error
    if (estimated_SE_p_hat_boot == 0) return(NA)  # Handle division by zero

    # Calculate t-statistic
    p_boot_t <- (p_hat_boot - p_hat) / estimated_SE_p_hat_boot
    return(p_boot_t)
  })

  # Calculate the t-bootstrap intervals using quantiles
  boot_t_interval <- c(p_hat - quantile(boot_estimates, 1 - alpha / 2, na.rm = TRUE) * sd(boot_estimates
                       p_hat - quantile(boot_estimates, alpha / 2, na.rm = TRUE) * sd(boot_estimates))

  # Return both raw percentile and t-bootstrap intervals
  return(c(raw_interval, boot_t_interval))
  }
}
```

Creation of data

We generated $N = 1000$ random samples from a binomial where sample size n varies across 15, 30, to 100 and true proportion p varies from 0.01 to 0.99 (15 total values of p).

```r
# Set values
N <- 1000 # Number of random samples
n <- c(15, 30, 100) # Values of n
p <- seq(0.01, 0.99, length.out = 15) # Values of p
B <- 100 # Number of bootstrap resamples

# Function to generate random binomial sample data for each combination of n and p
GetData <- function(n_vals, p_vals, N = 1000) {
  sample_data <- list()  # Initialize a list to store sample data
  for (n in n_vals) {
    for (p in p_vals) {
      samples <- rbinom(N, size = n, prob = p)  # Generate N binomial samples
      col_name <- paste0("n", n, "p", p)  # Create column name for the dataset
      sample_data[[col_name]] <- samples  # Store samples in the list
    }
  }
  return(sample_data)
}
sample_data <- GetData(n, p, N)
```

Execute the Monte Carlo simulation

```r
# Load necessary libraries
library(tidyverse)
library(plotrix)
library(ggplot2)
library(gridExtra)  # For arranging multiple ggplots
```

First, we created several functions to get different piece of the results:

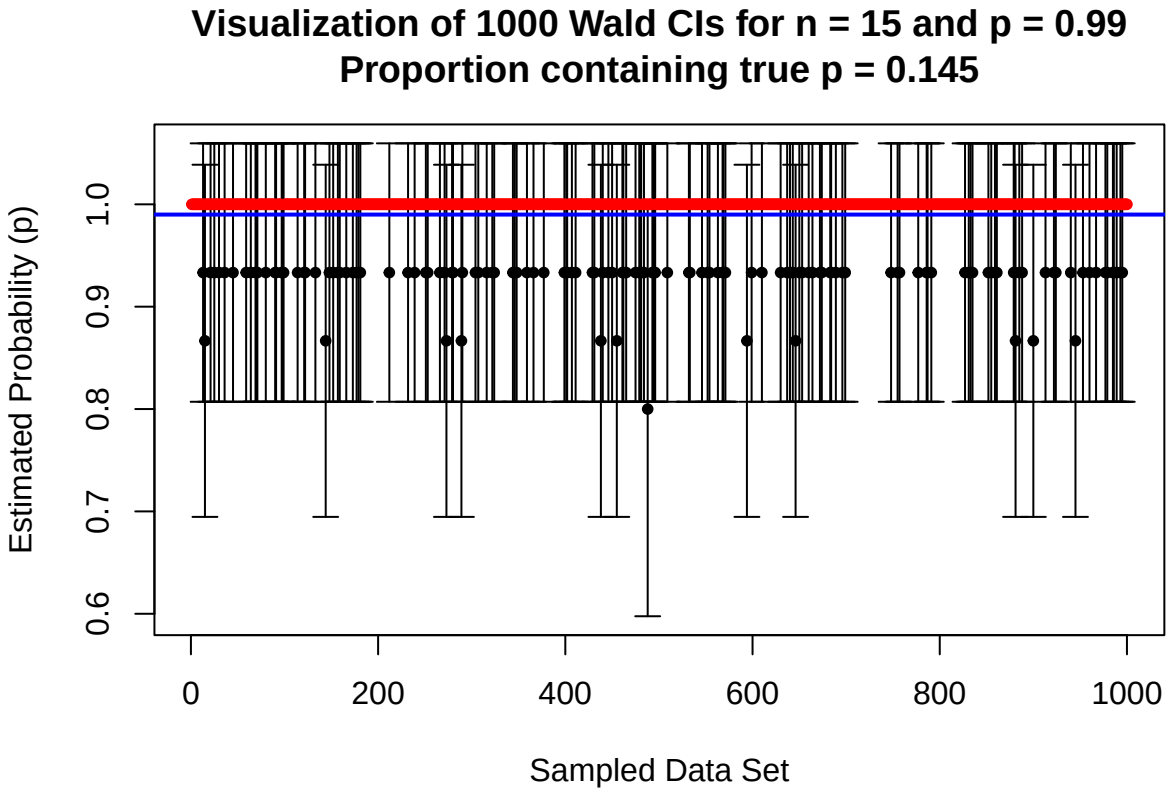
```r
# Function to calculate metrics such as coverage probability, average length, and errors
calculate_metrics <- function(cis, mu, ci_method, n, p) {
  probability <- mean(cis[, 1] <= mu & cis[, 2] >= mu)  # Coverage probability
  avg_length <- mean(cis[, 2] - cis[, 1])  # Average length of confidence intervals
  ci_length <- cis[, 2] - cis[, 1]  # Calculate individual lengths
  se_value <- sd(ci_length) / sqrt(length(ci_length))  # Standard error of lengths
  se_length <- round(se_value, 4)  # Round to 4 decimals
  err_below <- mean(cis[, 2] < mu)  # Error when CI is below true value
  err_above <- mean(cis[, 1] > mu)  # Error when CI is above true value

  # Return a data frame with the metrics
  data.frame(
    n = n,
    p = p,
    ci_method = ci_method,
    prob = probability,
    avg_length = avg_length,
    se_length = se_length,
    err_below = err_below,
    err_above = err_above
  )
```

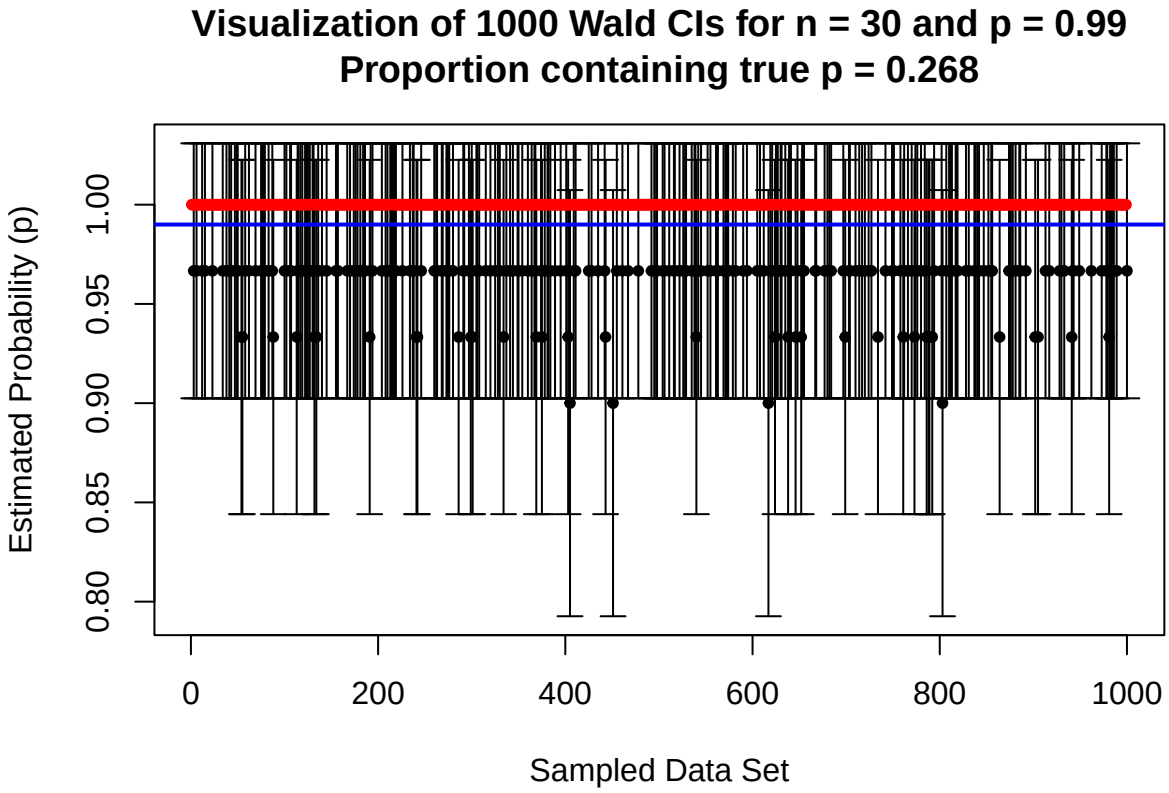
```r
}

# Function to assign color based on whether the true parameter is within the CI
mycolor <- function(endpoints, mu) {
  if (is.na(endpoints[1]) || is.na(endpoints[2])) {
    return("Black")  # Default color if CI endpoints are NA
  } else if (mu < endpoints[1]) {
    return("Red")  # True value is below the CI
  } else if (mu > endpoints[2]) {
    return("Orange")  # True value is above the CI
  } else {
    return("Black")  # True value lies within the CI
  }
}

# Function to plot the CIs for each method and sample
plot_CIs <- function(n, p, observed_CIs, ci_name, mu, N = 100) {
  # Plot the confidence intervals for N samples
  plotCI(x = 1:N,
         y = rowMeans(observed_CIs[1:N, ]),  # Mean of CIs for each sample
         li = observed_CIs[1:N, 1],  # Lower CI bound
         ui = observed_CIs[1:N, 2],  # Upper CI bound
         col = apply(FUN = mycolor, X = observed_CIs[1:N, ], MARGIN = 1, mu = mu),  # Assign color
         ylab = "Estimated Probability (p)",
         xlab = "Sampled Data Set",
         main = paste0("Visualization of 1000 ", ci_name, " CIs for n = ", n, " and p = ", p,
                       "\nProportion containing true p = ",
                       mean((observed_CIs[1:N, 1] < mu) &
                              (observed_CIs[1:N, 2] > mu))),
         pch = 20)

  abline(h = mu, lwd = 2, col = "blue")  # Draw a line for the true value of p
}

# Function to simulate and store metrics for different CI methods
simulate_CIs <- function(n_vals, p_vals, ci_function, ci_name, sample_data, N = 1000, alpha = 0.05) {
  results <- data.frame()
  coverage_results <- data.frame()  # To store the coverage probabilities
  # Simulate for each combination of n and p
  for (n in n_vals) {
    for (p in p_vals) {
      # Store the true probability for plotting
      mu <- p
      col_name <- paste0("n", n, "p", p)
      samples <- sample_data[[col_name]]

      # Compute confidence intervals for each sample
      #samples: This is a vector or list of binomial samples where each entry is a
      #     number of successes (y), given the sample size (n).
      #ci_function: This is the confidence interval function being applied (WaldCI, AdjWaldCI, etc.),wh
      #     the number of successes (y) and the sample size (n) as input to compute the confidence inte
      #sapply: This function applies ci_function to each element in samples, iterating over
      #     each sample (y) to compute the corresponding confidence interval.
```

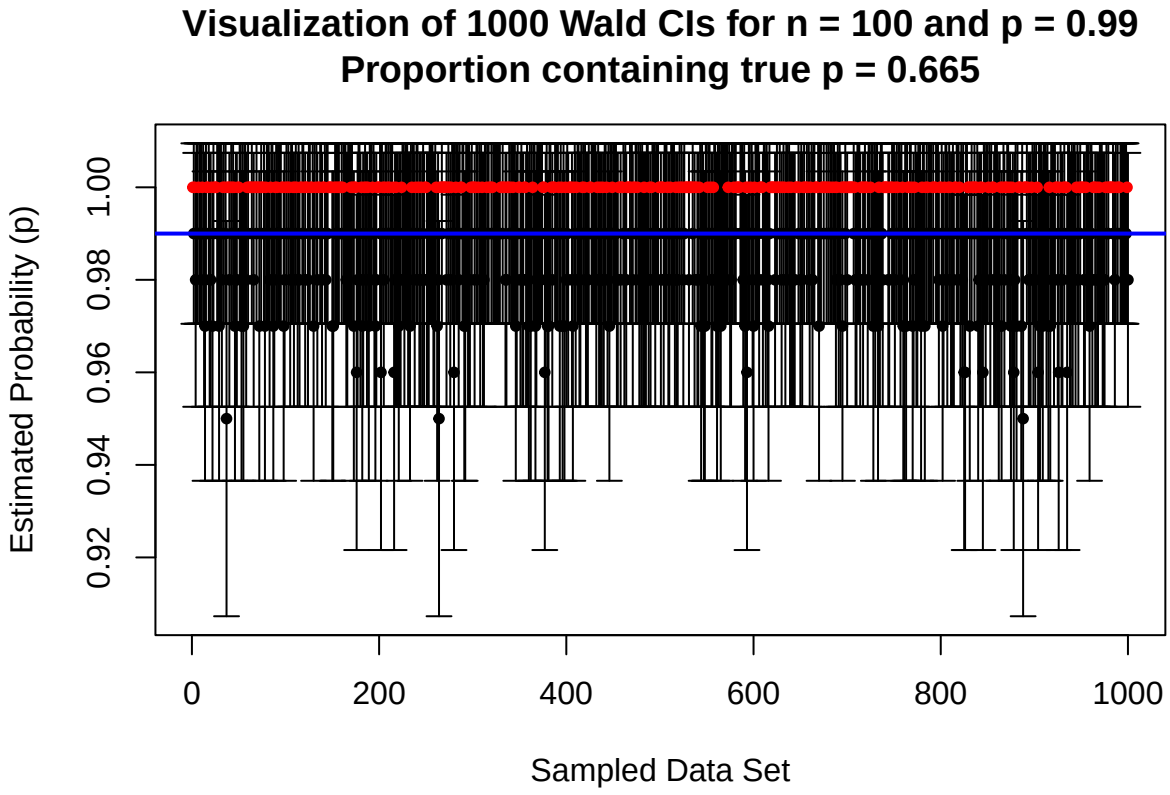
```
      #t: Transposes the result of sapply, ensuring that each computed confidence interval
      #     is placed in its own row, with the lower and upper bounds as separate columns.
      cis <- t(sapply(samples, function(y) ci_function(y, n)))

      # Calculate the metrics and create a data frame
      metrics_df <- calculate_metrics(cis, mu, ci_name, n, p)

      # Append the result in overall results data frame
      results <- rbind(results, metrics_df)

      # Plot the CIs
      if (p == 0.99) {
        plot_CIs(n, p, cis, ci_name, mu, N)
      }
    }
  }
  return(results)
}

simulate_BootstrapCIs <- function(n_vals, p_vals, ci_function, ci_name, sample_data, N = 1000 , B, alpha
  results <- data.frame()
  # Simulate for each combination of n and p
  for (n in n_vals) {
    for (p in p_vals) {
      mu <- p  # Store the true probability for plotting
      col_name <- paste0("n", n, "p", p)
      samples <- sample_data[[col_name]]  # Extract the sample data for this n and p

      # Compute confidence intervals using the BootstrapCI function
      cis <- t(sapply(samples, function(y) ci_function(y, n, size = n, B, alpha = alpha)))

      # Extract raw percentile and bootstrap t intervals from cis
      cis_raw <- cis[, 1:2]  # First two columns are raw_percentile
      cis_boot_t <- cis[, 3:4]  # Last two columns are boot_t_interval

      # Calculate the metrics for the percentile intervals
      raw_metrics <- calculate_metrics(cis_raw, mu, paste0(ci_name, " Raw"), n, p)
      results <- rbind(results, raw_metrics)

      # Calculate the metrics for the t intervals
      boot_t_metrics <- calculate_metrics(cis_boot_t, mu, paste0(ci_name, " t"), n, p)
      results <- rbind(results, boot_t_metrics)
    }
  }
  return(results)
}
```

Then we run the simulations:

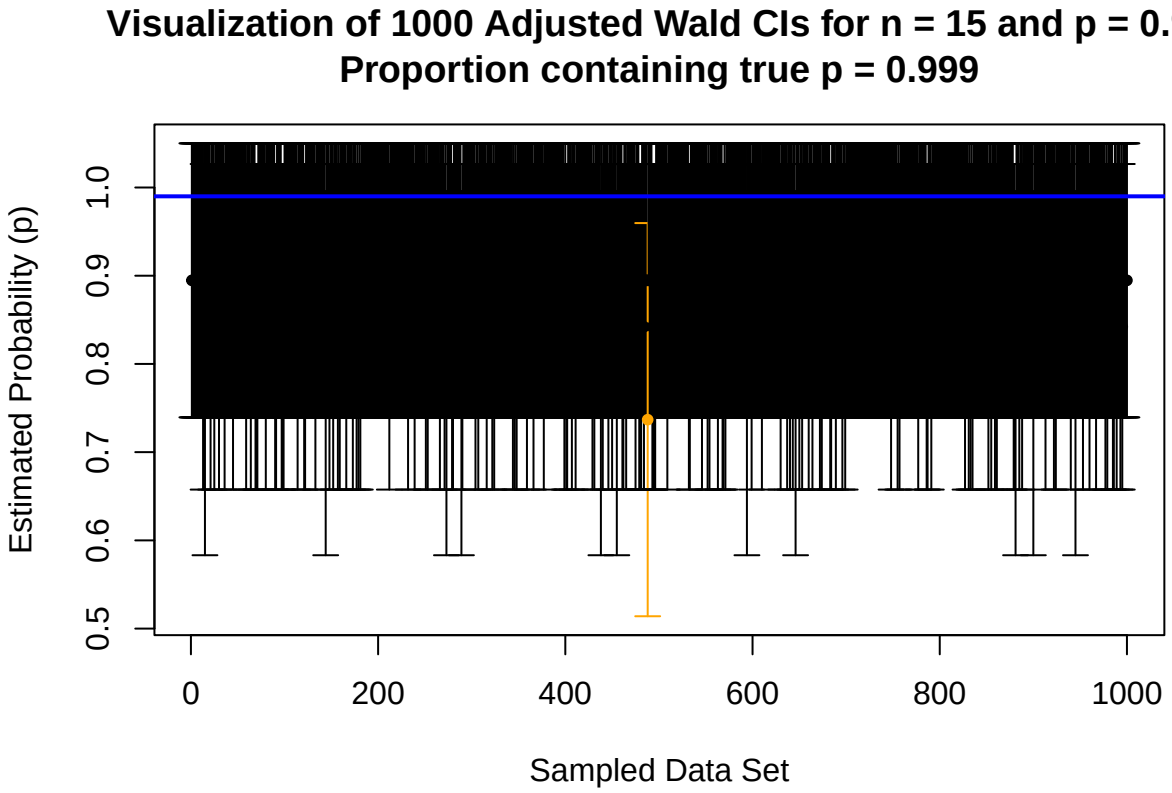
```
# Simulate CI intervals and metrices
wald_metrics <- simulate_CIs(n, p, WaldCI, "Wald", sample_data,N)
```

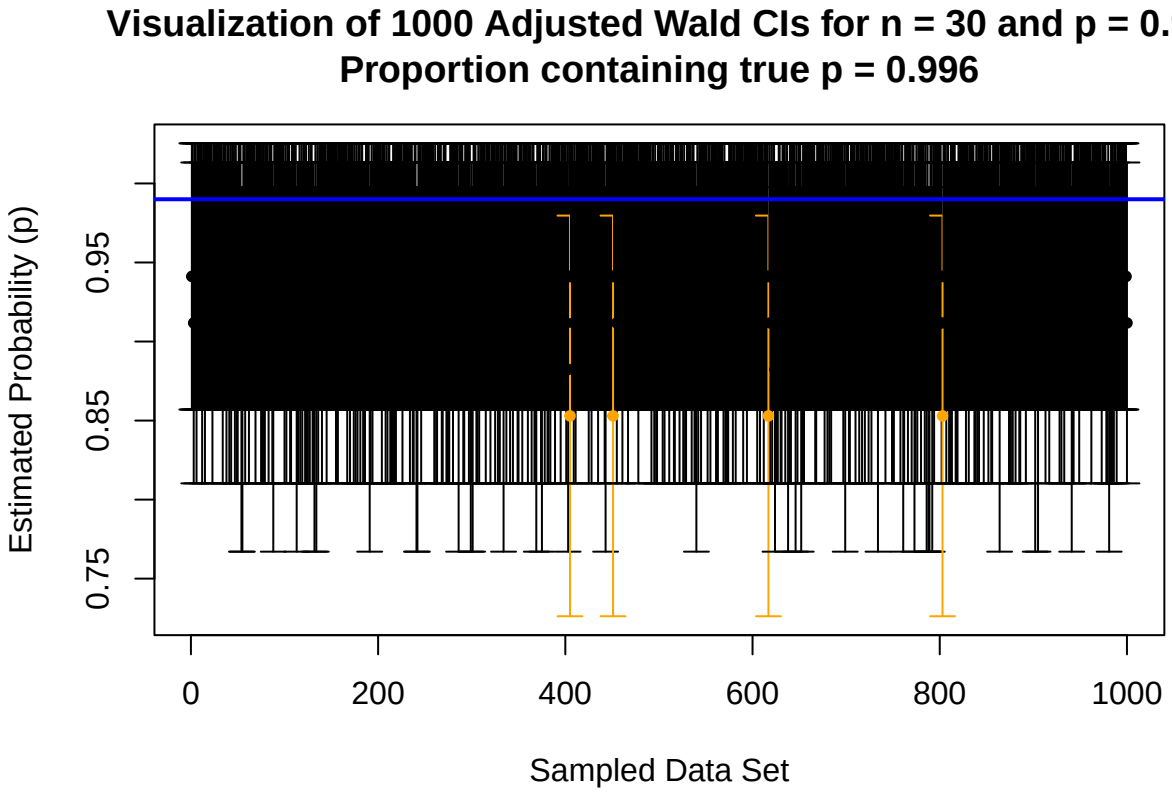

Visualization of 1000 Wald CIs for n = 15 and p = 0.99
Proportion containing true p = 0.145
Estimated Probability (p)
0.6
0.7
0.8
0.9
1.0
0
200
400
600
800
1000
Sampled Data Set

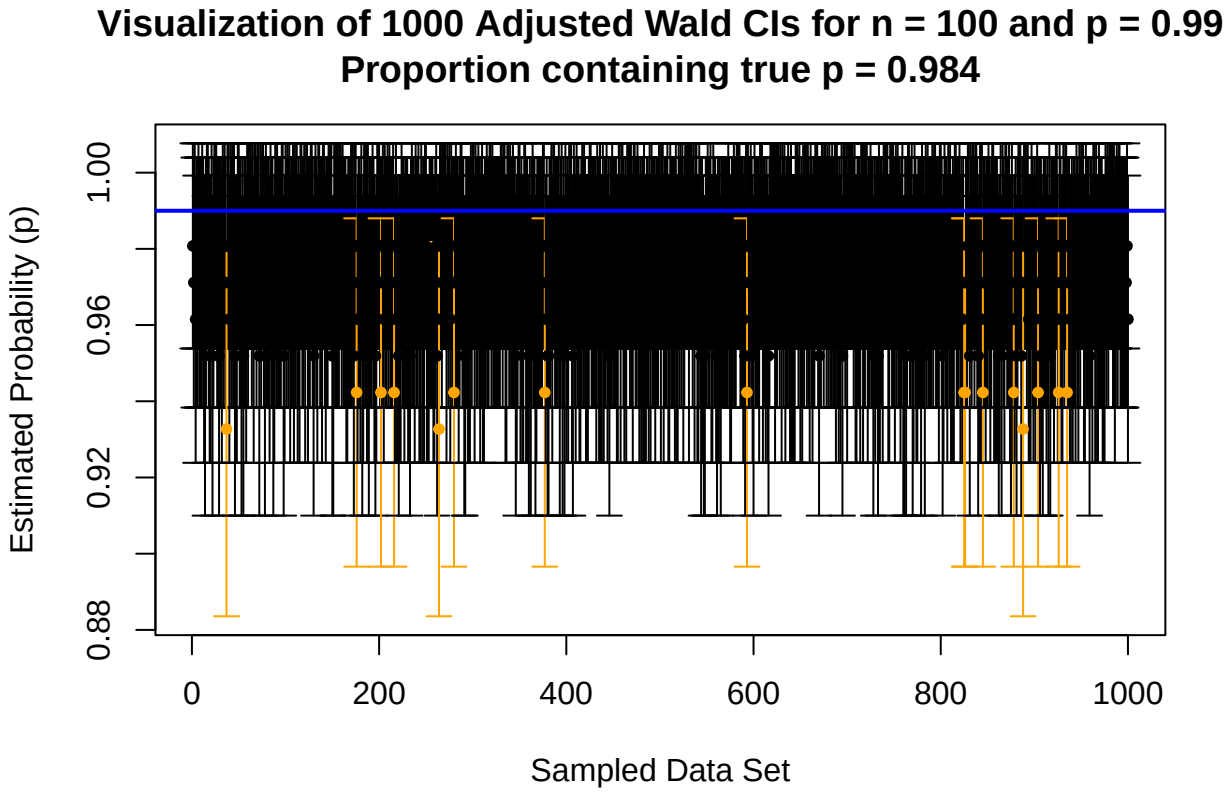

Visualization of 1000 Wald CIs for n = 30 and p = 0.99
Proportion containing true p = 0.268
Estimated Probability (p)
0.80
0.85
0.90
0.95
1.00
0
200
400
600
800
1000
Sampled Data Set



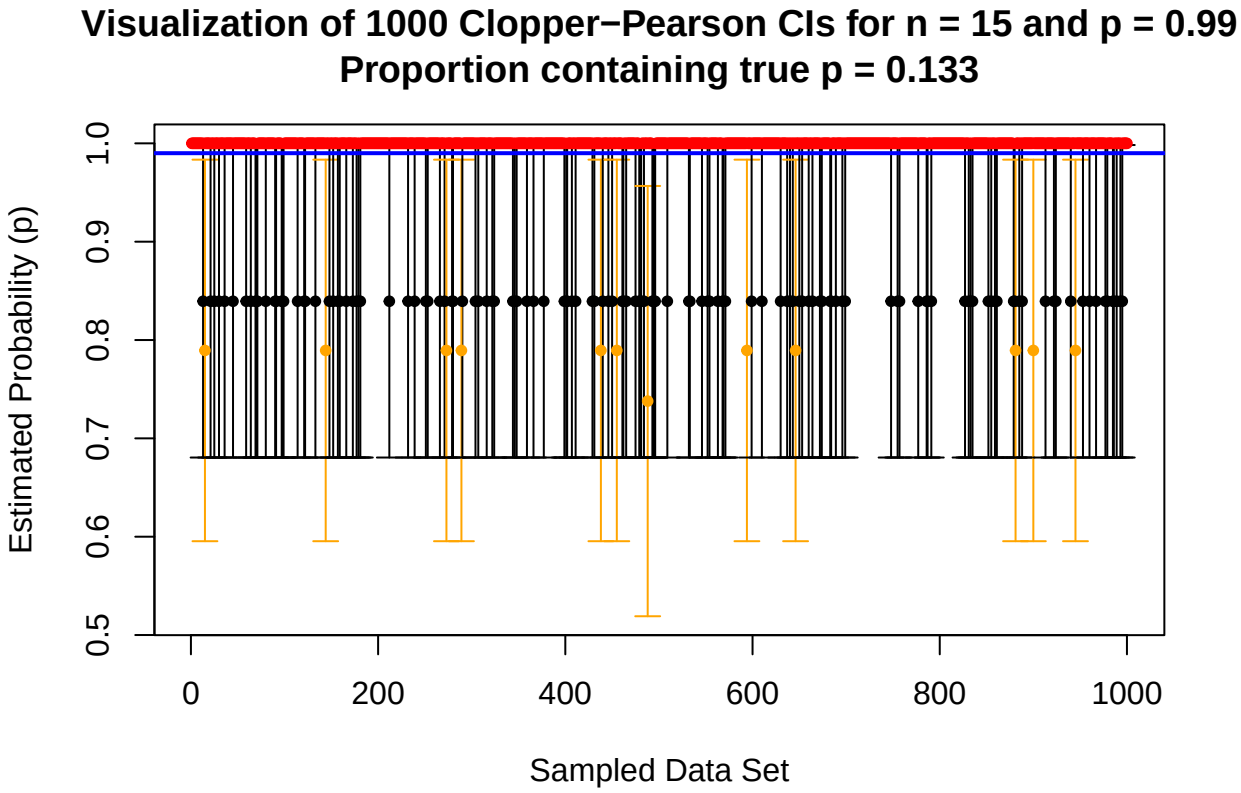

```
adj_wald_metrics <- simulate_CIs(n, p, AdjWaldCI, "Adjusted Wald", sample_data, N)
```

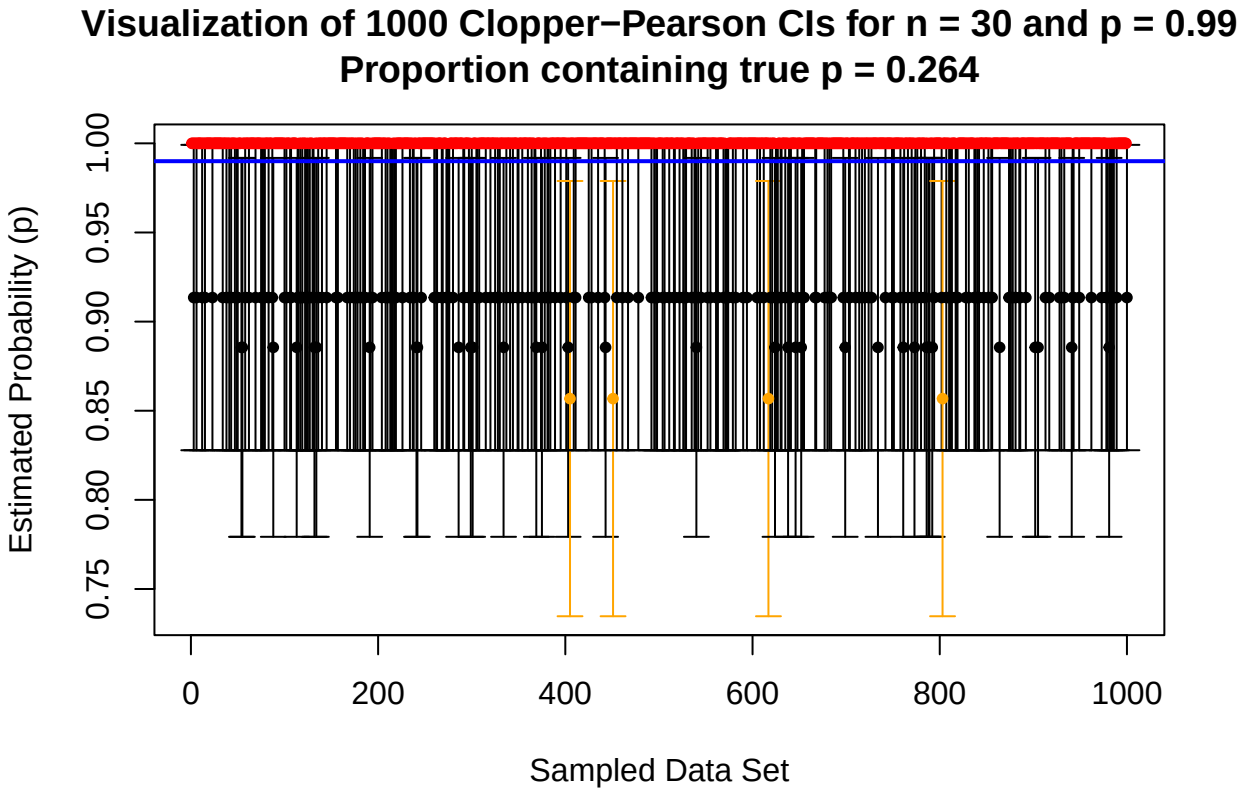

Visualization of 1000 Adjusted Wald CIs for n = 15 and p = 0.99
Proportion containing true p = 0.999
Estimated Probability (p)
0.5
0.6
0.7
0.8
0.9
1.0
0
200
400
600
800
1000
Sampled Data Set

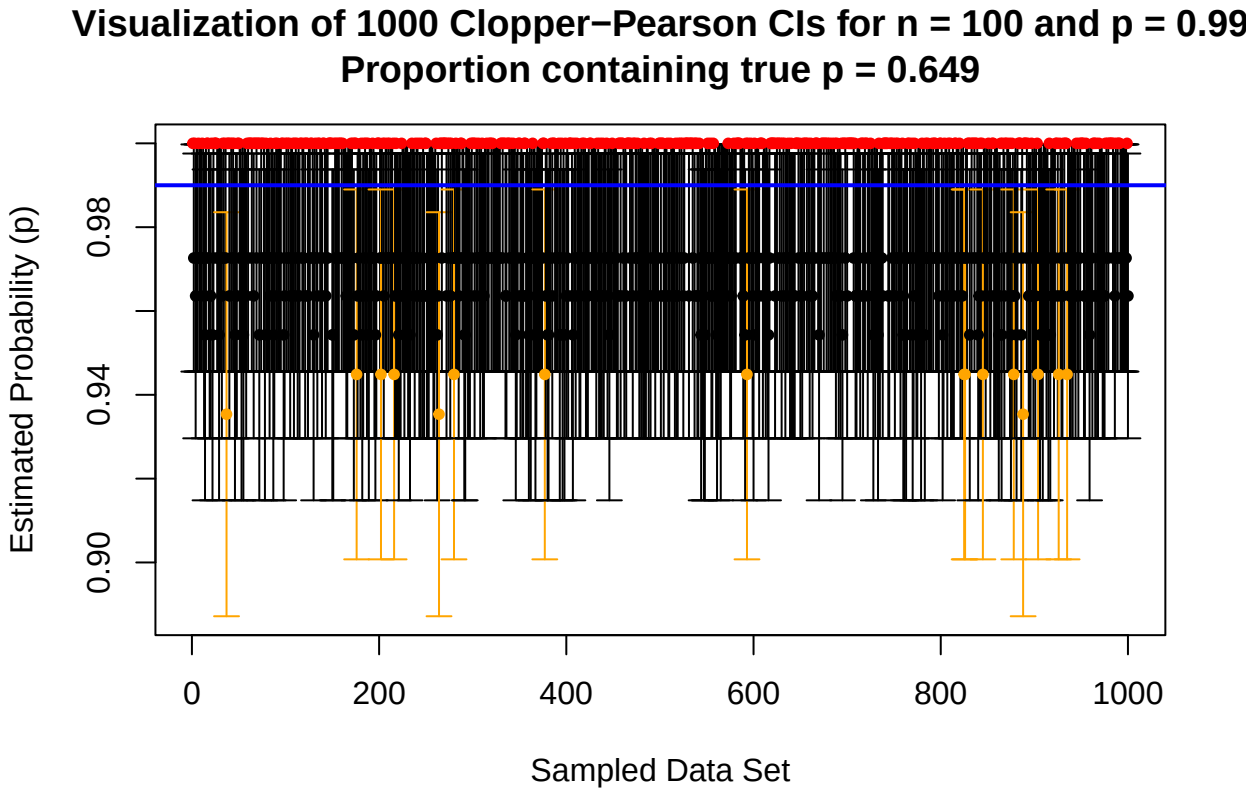

Visualization of 1000 Adjusted Wald CIs for n = 30 and p = 0.99
Proportion containing true p = 0.996
Estimated Probability (p)
0.75
0.85
0.95
0
200
400
600
800
1000
Sampled Data Set



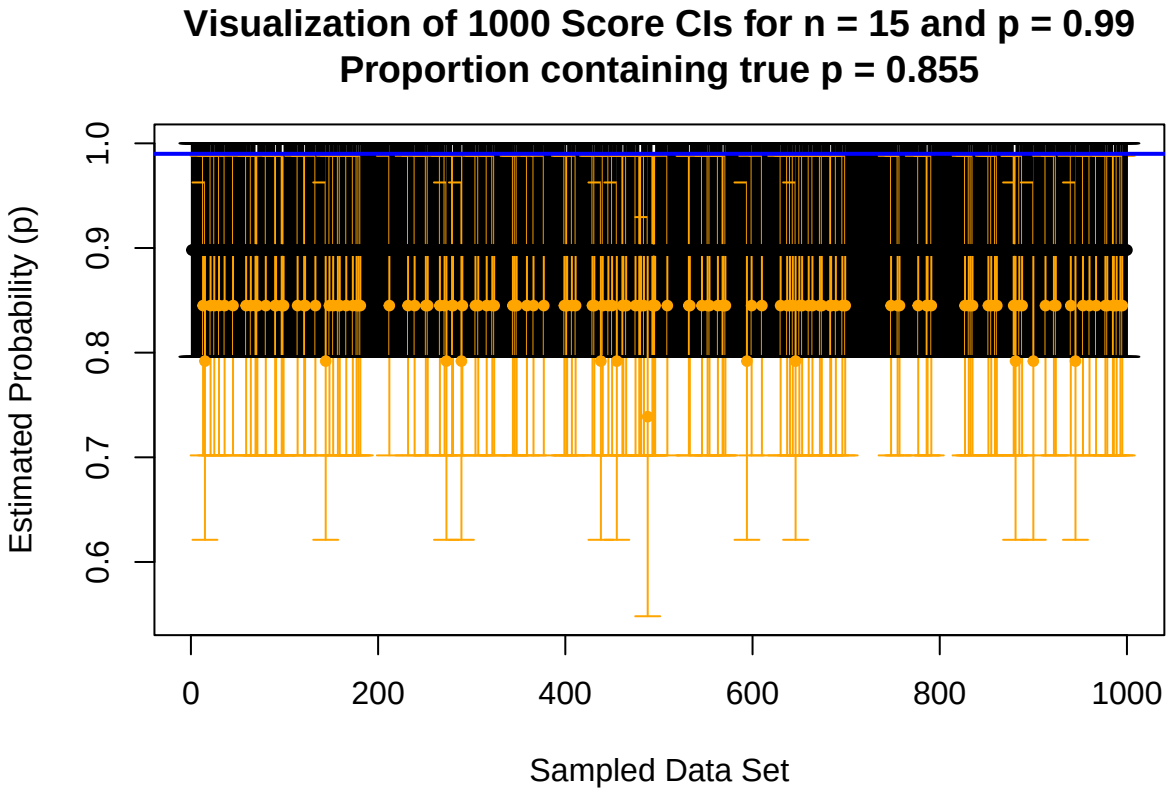

```
clopper_pearson_metrics <- simulate_CIs(n, p, ClopperPearsonCI, "Clopper-Pearson", sample_data, N)
```

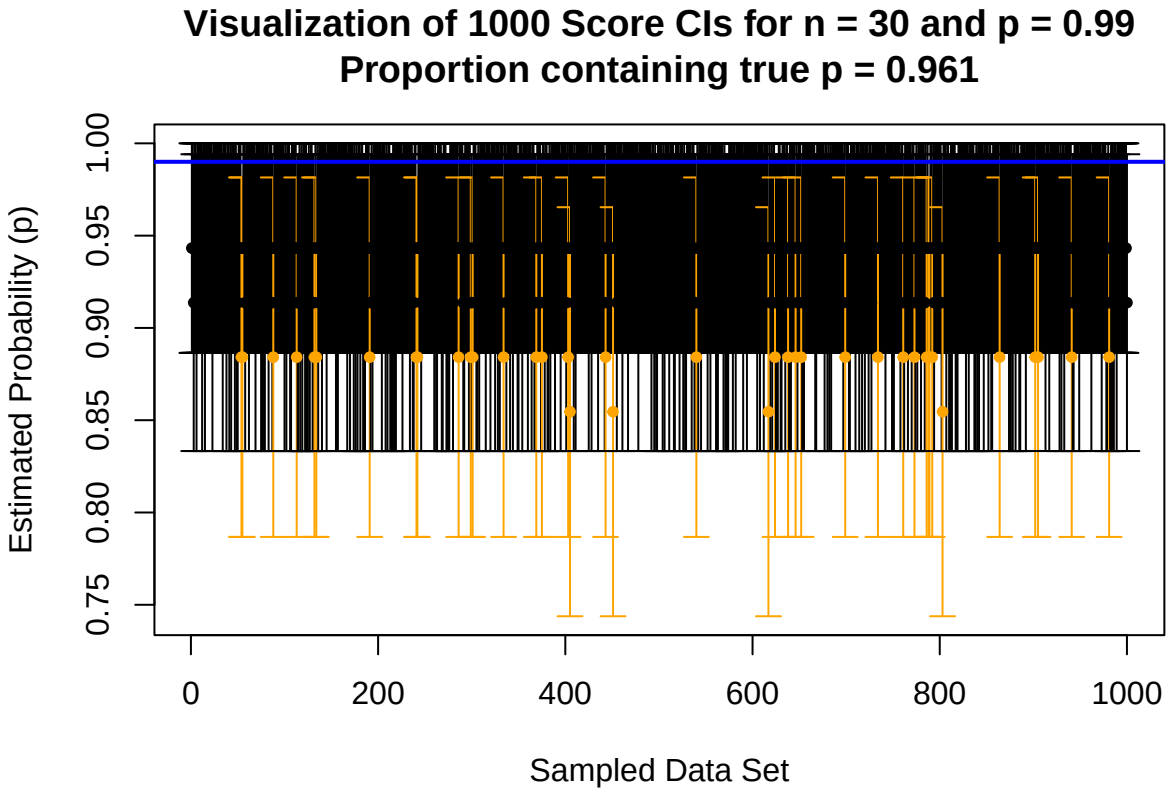

Visualization of 1000 Clopper−Pearson CIs for n = 15 and p = 0.99
Proportion containing true p = 0.133
Estimated Probability (p)
0.5
0.6
0.7
0.8
0.9
1.0
0
200
400
600
800
1000
Sampled Data Set

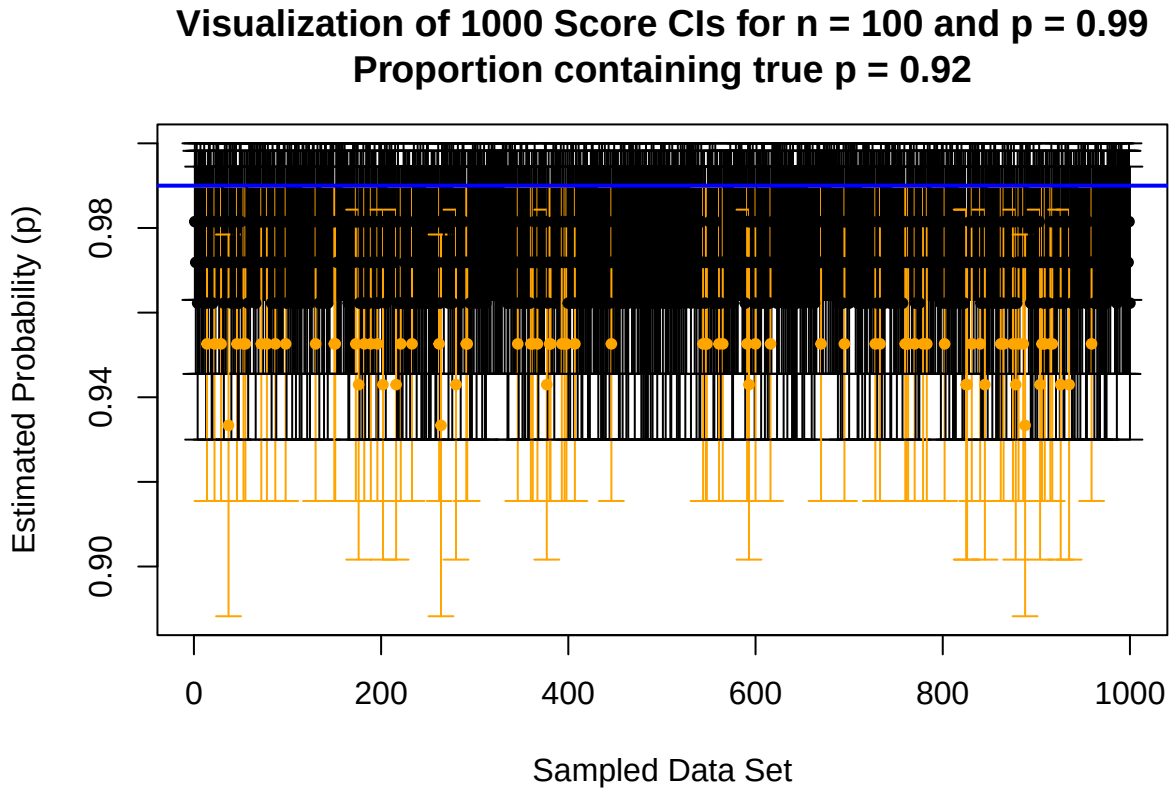

Visualization of 1000 Clopper−Pearson CIs for n = 30 and p = 0.99
Proportion containing true p = 0.264
Estimated Probability (p)
0.75
0.80
0.85
0.90
0.95
1.00
0
200
400
600
800
1000
Sampled Data Set




```
score_metrics <- simulate_CIs(n, p, ScoreCI, "Score", sample_data, N)
```


Visualization of 1000 Score CIs for n = 15 and p = 0.99
Proportion containing true p = 0.855
Estimated Probability (p)
1.0
0.9
0.8
0.7
0.6
0
200
400
600
800
1000
Sampled Data Set


Visualization of 1000 Score CIs for n = 30 and p = 0.99
Proportion containing true p = 0.961
Estimated Probability (p)
1.00
0.95
0.90
0.85
0.80
0.75
0
200
400
600
800
1000
Sampled Data Set




```
bootstrap_metrics <- simulate_BootstrapCIs(n , p, BootstrapCI, "Bootstrap", sample_data, N, B)

# Filter out metrics specifically for the Bootstrap t-interval and Bootstrap raw percentile method
bootstrap_t_metrics <- bootstrap_metrics %>% filter(ci_method == "Bootstrap t")
bootstrap_raw_metrics <- bootstrap_metrics %>% filter(ci_method == "Bootstrap Raw")

# Combine all results
all_metrics_combined <- rbind(wald_metrics, adj_wald_metrics, clopper_pearson_metrics, score_metrics, bo

# Sort the combined data frame by ci_method
all_metrics_combined <- all_metrics_combined[order(all_metrics_combined$ci_method), ]
print(all_metrics_combined)
```

```
##       n    p       ci_method  prob avg_length se_length err_below err_above
## 46   15 0.01   Adjusted Wald 0.999 0.32041786    0.0007     0.000     0.001
## 47   15 0.08   Adjusted Wald 0.966 0.37172015    0.0015     0.000     0.034
## 48   15 0.15   Adjusted Wald 0.984 0.41169358    0.0015     0.000     0.016
## 49   15 0.22   Adjusted Wald 0.970 0.44540088    0.0014     0.000     0.030
## 50   15 0.29   Adjusted Wald 0.980 0.46819698    0.0011     0.006     0.014
## 51   15 0.36   Adjusted Wald 0.976 0.48184012    0.0009     0.012     0.012
## 52   15 0.43   Adjusted Wald 0.969 0.49270576    0.0006     0.013     0.018
## 53   15 0.50   Adjusted Wald 0.968 0.49579496    0.0005     0.013     0.019
## 54   15 0.57   Adjusted Wald 0.968 0.49193460    0.0006     0.017     0.015
## 55   15 0.64   Adjusted Wald 0.984 0.48275316    0.0008     0.005     0.011
## 56   15 0.71   Adjusted Wald 0.985 0.46844326    0.0011     0.014     0.001
```

```
## 57   15 0.78   Adjusted Wald 0.970 0.44472169    0.0014     0.030     0.000
## 58   15 0.85   Adjusted Wald 0.984 0.41042717    0.0015     0.016     0.000
## 59   15 0.92   Adjusted Wald 0.979 0.36920348    0.0015     0.021     0.000
## 60   15 0.99   Adjusted Wald 0.999 0.31964312    0.0007     0.001     0.000
## 61   30 0.01   Adjusted Wald 0.997 0.17880509    0.0006     0.000     0.003
## 62   30 0.08   Adjusted Wald 0.977 0.23564089    0.0011     0.000     0.023
## 63   30 0.15   Adjusted Wald 0.967 0.27590973    0.0011     0.011     0.022
## 64   30 0.22   Adjusted Wald 0.971 0.30607042    0.0009     0.005     0.024
## 65   30 0.29   Adjusted Wald 0.962 0.32728408    0.0007     0.013     0.025
## 66   30 0.36   Adjusted Wald 0.961 0.34215588    0.0005     0.018     0.021
## 67   30 0.43   Adjusted Wald 0.964 0.35031794    0.0003     0.022     0.014
## 68   30 0.50   Adjusted Wald 0.960 0.35325268    0.0002     0.024     0.016
## 69   30 0.57   Adjusted Wald 0.962 0.35015330    0.0003     0.019     0.019
## 70   30 0.64   Adjusted Wald 0.971 0.34203914    0.0005     0.019     0.010
## 71   30 0.71   Adjusted Wald 0.967 0.32713839    0.0007     0.023     0.010
## 72   30 0.78   Adjusted Wald 0.982 0.30422215    0.0009     0.018     0.000
## 73   30 0.85   Adjusted Wald 0.966 0.27523834    0.0011     0.019     0.015
## 74   30 0.92   Adjusted Wald 0.968 0.23538685    0.0011     0.032     0.000
## 75   30 0.99   Adjusted Wald 0.996 0.17883455    0.0006     0.004     0.000
## 76  100 0.01   Adjusted Wald 0.984 0.06416754    0.0003     0.000     0.016
## 77  100 0.08   Adjusted Wald 0.961 0.11353072    0.0005     0.010     0.029
## 78  100 0.15   Adjusted Wald 0.967 0.14407892    0.0004     0.010     0.023
## 79  100 0.22   Adjusted Wald 0.950 0.16362007    0.0003     0.019     0.031
## 80  100 0.29   Adjusted Wald 0.955 0.17793331    0.0002     0.020     0.025
## 81  100 0.36   Adjusted Wald 0.945 0.18811485    0.0002     0.026     0.029
## 82  100 0.43   Adjusted Wald 0.955 0.19339774    0.0001     0.020     0.025
## 83  100 0.50   Adjusted Wald 0.971 0.19509150    0.0000     0.014     0.015
## 84  100 0.57   Adjusted Wald 0.940 0.19326278    0.0001     0.028     0.032
## 85  100 0.64   Adjusted Wald 0.955 0.18769896    0.0002     0.017     0.028
## 86  100 0.71   Adjusted Wald 0.961 0.17863640    0.0002     0.028     0.011
## 87  100 0.78   Adjusted Wald 0.953 0.16400769    0.0003     0.028     0.019
## 88  100 0.85   Adjusted Wald 0.962 0.14350536    0.0004     0.022     0.016
## 89  100 0.92   Adjusted Wald 0.970 0.11370911    0.0004     0.024     0.006
## 90  100 0.99   Adjusted Wald 0.984 0.06524009    0.0003     0.016     0.000
## 226  15 0.01   Bootstrap Raw 0.000 0.00000000    0.0000     0.841     0.159
## 227  15 0.08   Bootstrap Raw 0.000 0.00000000    0.0000     0.661     0.339
## 228  15 0.15   Bootstrap Raw 0.000 0.00000000    0.0000     0.608     0.392
## 229  15 0.22   Bootstrap Raw 0.000 0.00000000    0.0000     0.548     0.452
## 230  15 0.29   Bootstrap Raw 0.000 0.00000000    0.0000     0.542     0.458
## 231  15 0.36   Bootstrap Raw 0.000 0.00000000    0.0000     0.569     0.431
## 232  15 0.43   Bootstrap Raw 0.000 0.00000000    0.0000     0.531     0.469
## 233  15 0.50   Bootstrap Raw 0.000 0.00000000    0.0000     0.497     0.503
## 234  15 0.57   Bootstrap Raw 0.000 0.00000000    0.0000     0.465     0.535
## 235  15 0.64   Bootstrap Raw 0.000 0.00000000    0.0000     0.453     0.547
## 236  15 0.71   Bootstrap Raw 0.000 0.00000000    0.0000     0.477     0.523
## 237  15 0.78   Bootstrap Raw 0.000 0.00000000    0.0000     0.429     0.571
## 238  15 0.85   Bootstrap Raw 0.000 0.00000000    0.0000     0.386     0.614
## 239  15 0.92   Bootstrap Raw 0.000 0.00000000    0.0000     0.317     0.683
## 240  15 0.99   Bootstrap Raw 0.000 0.00000000    0.0000     0.145     0.855
## 241  30 0.01   Bootstrap Raw 0.000 0.00000000    0.0000     0.729     0.271
## 242  30 0.08   Bootstrap Raw 0.000 0.00000000    0.0000     0.571     0.429
## 243  30 0.15   Bootstrap Raw 0.000 0.00000000    0.0000     0.531     0.469
## 244  30 0.22   Bootstrap Raw 0.000 0.00000000    0.0000     0.509     0.491
## 245  30 0.29   Bootstrap Raw 0.000 0.00000000    0.0000     0.471     0.529
```

```
## 246  30 0.36   Bootstrap Raw 0.000 0.00000000    0.0000     0.445     0.555
## 247  30 0.43   Bootstrap Raw 0.000 0.00000000    0.0000     0.437     0.563
## 248  30 0.50   Bootstrap Raw 0.000 0.00000000    0.0000     0.417     0.583
## 249  30 0.57   Bootstrap Raw 0.000 0.00000000    0.0000     0.545     0.455
## 250  30 0.64   Bootstrap Raw 0.000 0.00000000    0.0000     0.542     0.458
## 251  30 0.71   Bootstrap Raw 0.000 0.00000000    0.0000     0.518     0.482
## 252  30 0.78   Bootstrap Raw 0.000 0.00000000    0.0000     0.465     0.535
## 253  30 0.85   Bootstrap Raw 0.000 0.00000000    0.0000     0.472     0.528
## 254  30 0.92   Bootstrap Raw 0.000 0.00000000    0.0000     0.427     0.573
## 255  30 0.99   Bootstrap Raw 0.000 0.00000000    0.0000     0.268     0.732
## 256 100 0.01   Bootstrap Raw 0.371 0.00000000    0.0000     0.384     0.245
## 257 100 0.08   Bootstrap Raw 0.000 0.00000000    0.0000     0.467     0.533
## 258 100 0.15   Bootstrap Raw 0.116 0.00000000    0.0000     0.454     0.430
## 259 100 0.22   Bootstrap Raw 0.000 0.00000000    0.0000     0.481     0.519
## 260 100 0.29   Bootstrap Raw 0.088 0.00000000    0.0000     0.491     0.421
## 261 100 0.36   Bootstrap Raw 0.083 0.00000000    0.0000     0.427     0.490
## 262 100 0.43   Bootstrap Raw 0.000 0.00000000    0.0000     0.449     0.551
## 263 100 0.50   Bootstrap Raw 0.000 0.00000000    0.0000     0.476     0.524
## 264 100 0.57   Bootstrap Raw 0.081 0.00000000    0.0000     0.450     0.469
## 265 100 0.64   Bootstrap Raw 0.000 0.00000000    0.0000     0.454     0.546
## 266 100 0.71   Bootstrap Raw 0.098 0.00000000    0.0000     0.459     0.443
## 267 100 0.78   Bootstrap Raw 0.000 0.00000000    0.0000     0.449     0.551
## 268 100 0.85   Bootstrap Raw 0.000 0.00000000    0.0000     0.436     0.564
## 269 100 0.92   Bootstrap Raw 0.000 0.00000000    0.0000     0.385     0.615
## 270 100 0.99   Bootstrap Raw 0.383 0.00000000    0.0000     0.282     0.335
## 181  15 0.01     Bootstrap t 0.159 0.72360088    0.0539     0.841     0.000
## 182  15 0.08     Bootstrap t 0.717 3.10384621    0.0654     0.283     0.000
## 183  15 0.15     Bootstrap t 0.912 3.84351492    0.0445     0.088     0.000
## 184  15 0.22     Bootstrap t 0.971 3.97038627    0.0304     0.029     0.000
## 185  15 0.29     Bootstrap t 0.994 4.05326386    0.0238     0.006     0.000
## 186  15 0.36     Bootstrap t 0.998 4.00420206    0.0214     0.002     0.000
## 187  15 0.43     Bootstrap t 1.000 4.02519912    0.0201     0.000     0.000
## 188  15 0.50     Bootstrap t 1.000 3.98462544    0.0202     0.000     0.000
## 189  15 0.57     Bootstrap t 0.998 4.03120879    0.0215     0.000     0.002
## 190  15 0.64     Bootstrap t 1.000 3.98290680    0.0193     0.000     0.000
## 191  15 0.71     Bootstrap t 0.999 4.05049434    0.0209     0.000     0.001
## 192  15 0.78     Bootstrap t 0.980 4.00169157    0.0280     0.000     0.020
## 193  15 0.85     Bootstrap t 0.914 3.81141618    0.0433     0.000     0.086
## 194  15 0.92     Bootstrap t 0.695 3.02577921    0.0666     0.000     0.305
## 195  15 0.99     Bootstrap t 0.145 0.65634079    0.0517     0.000     0.855
## 196  30 0.01     Bootstrap t 0.271 1.12706413    0.0598     0.729     0.000
## 197  30 0.08     Bootstrap t 0.924 3.71959733    0.0393     0.076     0.000
## 198  30 0.15     Bootstrap t 0.989 3.97247740    0.0243     0.011     0.000
## 199  30 0.22     Bootstrap t 1.000 3.99264983    0.0210     0.000     0.000
## 200  30 0.29     Bootstrap t 1.000 3.95878405    0.0201     0.000     0.000
## 201  30 0.36     Bootstrap t 1.000 3.93675023    0.0197     0.000     0.000
## 202  30 0.43     Bootstrap t 1.000 3.94024868    0.0191     0.000     0.000
## 203  30 0.50     Bootstrap t 1.000 3.95797138    0.0204     0.000     0.000
## 204  30 0.57     Bootstrap t 1.000 3.95424394    0.0196     0.000     0.000
## 205  30 0.64     Bootstrap t 1.000 3.95177628    0.0191     0.000     0.000
## 206  30 0.71     Bootstrap t 1.000 3.98352557    0.0206     0.000     0.000
## 207  30 0.78     Bootstrap t 1.000 3.98794626    0.0204     0.000     0.000
## 208  30 0.85     Bootstrap t 0.985 3.92420290    0.0252     0.000     0.015
## 209  30 0.92     Bootstrap t 0.919 3.73756248    0.0407     0.000     0.081
```

```
## 210  30 0.99     Bootstrap t 0.268 1.11626415    0.0594     0.000     0.732
## 211 100 0.01     Bootstrap t 0.616 2.48332213    0.0642     0.384     0.000
## 212 100 0.08     Bootstrap t 0.998 3.95408802    0.0200     0.002     0.000
## 213 100 0.15     Bootstrap t 1.000 3.93284060    0.0195     0.000     0.000
## 214 100 0.22     Bootstrap t 1.000 3.93485833    0.0198     0.000     0.000
## 215 100 0.29     Bootstrap t 1.000 3.95771423    0.0197     0.000     0.000
## 216 100 0.36     Bootstrap t 1.000 3.94603012    0.0203     0.000     0.000
## 217 100 0.43     Bootstrap t 1.000 3.95570005    0.0198     0.000     0.000
## 218 100 0.50     Bootstrap t 1.000 3.93772098    0.0193     0.000     0.000
## 219 100 0.57     Bootstrap t 1.000 3.94575704    0.0191     0.000     0.000
## 220 100 0.64     Bootstrap t 1.000 3.97953528    0.0192     0.000     0.000
## 221 100 0.71     Bootstrap t 1.000 3.95404954    0.0197     0.000     0.000
## 222 100 0.78     Bootstrap t 1.000 3.90254637    0.0192     0.000     0.000
## 223 100 0.85     Bootstrap t 1.000 3.90154429    0.0193     0.000     0.000
## 224 100 0.92     Bootstrap t 1.000 3.96716991    0.0204     0.000     0.000
## 225 100 0.99     Bootstrap t 0.665 2.64130099    0.0617     0.000     0.335
## 91   15 0.01 Clopper-Pearson 0.148 0.05135197    0.0037     0.841     0.011
## 92   15 0.08 Clopper-Pearson 0.711 0.25824531    0.0053     0.283     0.006
## 93   15 0.15 Clopper-Pearson 0.896 0.36432285    0.0040     0.088     0.016
## 94   15 0.22 Clopper-Pearson 0.966 0.42857425    0.0029     0.029     0.005
## 95   15 0.29 Clopper-Pearson 0.980 0.46742627    0.0019     0.006     0.014
## 96   15 0.36 Clopper-Pearson 0.976 0.48799101    0.0014     0.012     0.012
## 97   15 0.43 Clopper-Pearson 0.969 0.50389741    0.0008     0.013     0.018
## 98   15 0.50 Clopper-Pearson 0.968 0.50823120    0.0006     0.013     0.019
## 99   15 0.57 Clopper-Pearson 0.968 0.50230257    0.0011     0.017     0.015
## 100  15 0.64 Clopper-Pearson 0.984 0.48974156    0.0012     0.005     0.011
## 101  15 0.71 Clopper-Pearson 0.985 0.46887485    0.0016     0.014     0.001
## 102  15 0.78 Clopper-Pearson 0.970 0.42966274    0.0027     0.010     0.020
## 103  15 0.85 Clopper-Pearson 0.898 0.36285101    0.0040     0.016     0.086
## 104  15 0.92 Clopper-Pearson 0.693 0.24932964    0.0054     0.002     0.305
## 105  15 0.99 Clopper-Pearson 0.133 0.04697303    0.0036     0.012     0.855
## 106  30 0.01 Clopper-Pearson 0.268 0.04796726    0.0025     0.729     0.003
## 107  30 0.08 Clopper-Pearson 0.917 0.20869949    0.0023     0.076     0.007
## 108  30 0.15 Clopper-Pearson 0.979 0.27101969    0.0016     0.011     0.010
## 109  30 0.22 Clopper-Pearson 0.971 0.31101024    0.0011     0.005     0.024
## 110  30 0.29 Clopper-Pearson 0.976 0.33724823    0.0009     0.013     0.011
## 111  30 0.36 Clopper-Pearson 0.961 0.35529077    0.0006     0.018     0.021
## 112  30 0.43 Clopper-Pearson 0.964 0.36509960    0.0004     0.022     0.014
## 113  30 0.50 Clopper-Pearson 0.960 0.36861063    0.0002     0.024     0.016
## 114  30 0.57 Clopper-Pearson 0.962 0.36490736    0.0004     0.019     0.019
## 115  30 0.64 Clopper-Pearson 0.971 0.35516537    0.0006     0.019     0.010
## 116  30 0.71 Clopper-Pearson 0.984 0.33708241    0.0008     0.006     0.010
## 117  30 0.78 Clopper-Pearson 0.982 0.30870096    0.0011     0.018     0.000
## 118  30 0.85 Clopper-Pearson 0.978 0.26961883    0.0017     0.007     0.015
## 119  30 0.92 Clopper-Pearson 0.915 0.20767734    0.0023     0.004     0.081
## 120  30 0.99 Clopper-Pearson 0.264 0.04764982    0.0025     0.004     0.732
## 121 100 0.01 Clopper-Pearson 0.600 0.03763868    0.0010     0.384     0.016
## 122 100 0.08 Clopper-Pearson 0.976 0.11372694    0.0005     0.010     0.014
## 123 100 0.15 Clopper-Pearson 0.967 0.14782247    0.0004     0.010     0.023
## 124 100 0.22 Clopper-Pearson 0.950 0.16893216    0.0004     0.019     0.031
## 125 100 0.29 Clopper-Pearson 0.955 0.18422094    0.0003     0.020     0.025
## 126 100 0.36 Clopper-Pearson 0.954 0.19502563    0.0002     0.026     0.020
## 127 100 0.43 Clopper-Pearson 0.977 0.20061379    0.0001     0.008     0.015
## 128 100 0.50 Clopper-Pearson 0.971 0.20240271    0.0000     0.014     0.015
```

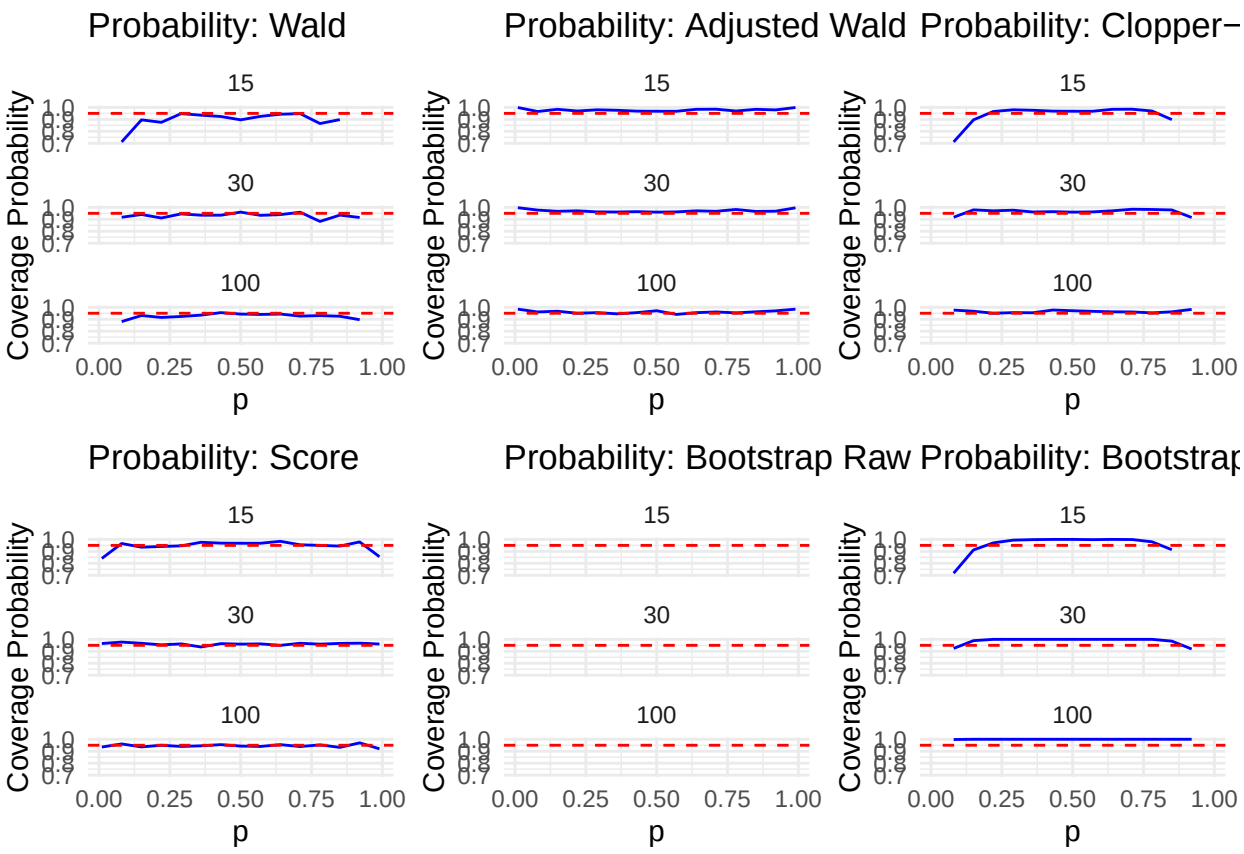
```
## 129 100 0.57 Clopper-Pearson 0.966 0.20047100    0.0001     0.015     0.019
## 130 100 0.64 Clopper-Pearson 0.962 0.19458551    0.0002     0.010     0.028
## 131 100 0.71 Clopper-Pearson 0.961 0.18496932    0.0003     0.028     0.011
## 132 100 0.78 Clopper-Pearson 0.953 0.16935032    0.0004     0.028     0.019
## 133 100 0.85 Clopper-Pearson 0.962 0.14718997    0.0004     0.022     0.016
## 134 100 0.92 Clopper-Pearson 0.982 0.11405010    0.0005     0.012     0.006
## 135 100 0.99 Clopper-Pearson 0.649 0.04097646    0.0010     0.016     0.335
## 136  15 0.01           Score 0.841 0.21763398    0.0010     0.000     0.159
## 137  15 0.08           Score 0.966 0.28699361    0.0020     0.000     0.034
## 138  15 0.15           Score 0.935 0.33824298    0.0020     0.000     0.065
## 139  15 0.22           Score 0.941 0.37974874    0.0017     0.029     0.030
## 140  15 0.29           Score 0.946 0.40726355    0.0013     0.006     0.048
## 141  15 0.36           Score 0.976 0.42339740    0.0010     0.012     0.012
## 142  15 0.43           Score 0.969 0.43613421    0.0007     0.013     0.018
## 143  15 0.50           Score 0.968 0.43971989    0.0005     0.013     0.019
## 144  15 0.57           Score 0.968 0.43518916    0.0008     0.017     0.015
## 145  15 0.64           Score 0.984 0.42448555    0.0010     0.005     0.011
## 146  15 0.71           Score 0.955 0.40758528    0.0013     0.044     0.001
## 147  15 0.78           Score 0.950 0.37905119    0.0017     0.030     0.020
## 148  15 0.85           Score 0.944 0.33671848    0.0019     0.056     0.000
## 149  15 0.92           Score 0.979 0.28365221    0.0020     0.021     0.000
## 150  15 0.99           Score 0.855 0.21653533    0.0010     0.145     0.000
## 151  30 0.01           Score 0.965 0.12759603    0.0008     0.000     0.035
## 152  30 0.08           Score 0.977 0.19945121    0.0013     0.000     0.023
## 153  30 0.15           Score 0.967 0.24648596    0.0012     0.011     0.022
## 154  30 0.22           Score 0.954 0.28043932    0.0010     0.022     0.024
## 155  30 0.29           Score 0.962 0.30379288    0.0008     0.013     0.025
## 156  30 0.36           Score 0.936 0.31998219    0.0005     0.018     0.046
## 157  30 0.43           Score 0.964 0.32881782    0.0003     0.022     0.014
## 158  30 0.50           Score 0.960 0.33198639    0.0002     0.024     0.016
## 159  30 0.57           Score 0.962 0.32864261    0.0003     0.019     0.019
## 160  30 0.64           Score 0.950 0.31986360    0.0005     0.040     0.010
## 161  30 0.71           Score 0.967 0.30364014    0.0007     0.023     0.010
## 162  30 0.78           Score 0.960 0.27839221    0.0010     0.018     0.022
## 163  30 0.85           Score 0.966 0.24568175    0.0012     0.019     0.015
## 164  30 0.92           Score 0.968 0.19909277    0.0013     0.032     0.000
## 165  30 0.99           Score 0.961 0.12761689    0.0008     0.039     0.000
## 166 100 0.01           Score 0.935 0.05035323    0.0004     0.000     0.065
## 167 100 0.08           Score 0.961 0.10656653    0.0005     0.010     0.029
## 168 100 0.15           Score 0.936 0.13881967    0.0004     0.027     0.037
## 169 100 0.22           Score 0.950 0.15908335    0.0004     0.019     0.031
## 170 100 0.29           Score 0.940 0.17382531    0.0003     0.028     0.032
## 171 100 0.36           Score 0.945 0.18427028    0.0002     0.026     0.029
## 172 100 0.43           Score 0.955 0.18967919    0.0001     0.020     0.025
## 173 100 0.50           Score 0.943 0.19141174    0.0000     0.027     0.030
## 174 100 0.57           Score 0.940 0.18954099    0.0001     0.028     0.032
## 175 100 0.64           Score 0.955 0.18384435    0.0002     0.017     0.028
## 176 100 0.71           Score 0.939 0.17454792    0.0002     0.038     0.023
## 177 100 0.78           Score 0.953 0.15948498    0.0003     0.028     0.019
## 178 100 0.85           Score 0.932 0.13821758    0.0004     0.036     0.032
## 179 100 0.92           Score 0.970 0.10678384    0.0005     0.024     0.006
## 180 100 0.99           Score 0.920 0.05171386    0.0004     0.080     0.000
## 1    15 0.01            Wald 0.159 0.04121057    0.0030     0.841     0.000
## 2    15 0.08            Wald 0.711 0.22009934    0.0047     0.283     0.006
```

```
## 3    15 0.15           Wald 0.896 0.32504420    0.0039     0.088     0.016
## 4    15 0.22           Wald 0.875 0.39643498    0.0031     0.120     0.005
## 5    15 0.29           Wald 0.948 0.44085723    0.0021     0.038     0.014
## 6    15 0.36           Wald 0.935 0.46520004    0.0016     0.053     0.012
## 7    15 0.43           Wald 0.924 0.48405292    0.0009     0.058     0.018
## 8    15 0.50           Wald 0.896 0.48921817    0.0008     0.053     0.051
## 9    15 0.57           Wald 0.925 0.48236988    0.0012     0.017     0.058
## 10   15 0.64           Wald 0.943 0.46703741    0.0015     0.005     0.052
## 11   15 0.71           Wald 0.948 0.44191191    0.0020     0.014     0.038
## 12   15 0.78           Wald 0.865 0.39653894    0.0029     0.010     0.125
## 13   15 0.85           Wald 0.898 0.32290376    0.0039     0.016     0.086
## 14   15 0.92           Wald 0.693 0.21208095    0.0047     0.002     0.305
## 15   15 0.99           Wald 0.145 0.03776761    0.0029     0.000     0.855
## 16   30 0.01           Wald 0.271 0.03667526    0.0019     0.729     0.000
## 17   30 0.08           Wald 0.917 0.17814166    0.0022     0.076     0.007
## 18   30 0.15           Wald 0.940 0.24489851    0.0017     0.050     0.010
## 19   30 0.22           Wald 0.911 0.28878282    0.0012     0.065     0.024
## 20   30 0.29           Wald 0.945 0.31765430    0.0009     0.044     0.011
## 21   30 0.36           Wald 0.934 0.33738651    0.0007     0.045     0.021
## 22   30 0.43           Wald 0.934 0.34808685    0.0004     0.052     0.014
## 23   30 0.50           Wald 0.960 0.35191231    0.0003     0.024     0.016
## 24   30 0.57           Wald 0.933 0.34787870    0.0004     0.019     0.048
## 25   30 0.64           Wald 0.939 0.33725438    0.0006     0.019     0.042
## 26   30 0.71           Wald 0.959 0.31747535    0.0009     0.006     0.035
## 27   30 0.78           Wald 0.882 0.28624696    0.0012     0.018     0.100
## 28   30 0.85           Wald 0.935 0.24355912    0.0017     0.007     0.058
## 29   30 0.92           Wald 0.915 0.17727355    0.0022     0.004     0.081
## 30   30 0.99           Wald 0.268 0.03652625    0.0019     0.000     0.732
## 31  100 0.01           Wald 0.616 0.02887110    0.0008     0.384     0.000
## 32  100 0.08           Wald 0.880 0.10352702    0.0006     0.106     0.014
## 33  100 0.15           Wald 0.931 0.13888741    0.0004     0.055     0.014
## 34  100 0.22           Wald 0.915 0.16064110    0.0004     0.064     0.021
## 35  100 0.29           Wald 0.923 0.17635778    0.0003     0.052     0.025
## 36  100 0.36           Wald 0.934 0.18744961    0.0002     0.037     0.029
## 37  100 0.43           Wald 0.955 0.19318244    0.0001     0.020     0.025
## 38  100 0.50           Wald 0.943 0.19501710    0.0000     0.027     0.030
## 39  100 0.57           Wald 0.940 0.19303595    0.0001     0.028     0.032
## 40  100 0.64           Wald 0.943 0.18699805    0.0002     0.017     0.040
## 41  100 0.71           Wald 0.926 0.17712654    0.0003     0.028     0.046
## 42  100 0.78           Wald 0.930 0.16107188    0.0004     0.013     0.057
## 43  100 0.85           Wald 0.925 0.13823253    0.0005     0.015     0.060
## 44  100 0.92           Wald 0.896 0.10384381    0.0005     0.012     0.092
## 45  100 0.99           Wald 0.665 0.03155813    0.0008     0.000     0.335
```

```
# Create individual plots for each CI method and value of n
plot_probability <- function(metrics_data, method_name) {
  ggplot(metrics_data %>% filter(ci_method == method_name),
         aes(x = p, y = prob)) +
    # Plot the coverage probability as a line plot
    geom_line(color = "blue") +
    facet_wrap(~n, ncol = 1) + # Create separate panels (facets) for each value of 'n'
    # Set plot labels including title, x-axis, and y-axis
    labs(
      title = paste("Probability:", method_name),
```

```
      x = "p", # X-axis label (values of 'p')
      y = "Coverage Probability" # Y-axis label
    ) +
    # Add a reference horizontal line at 0.95 to indicate the nominal coverage level
    geom_hline(yintercept = 0.95, linetype = "dashed", color = "red") +  # Add 95% reference line
    theme_minimal() +  # Minimal theme for clean look
    # Set y-axis limits to maintain consistency across plots
    ylim(0.7, 1.0)  # Coverage probabilities between 0.7 and 1.0
}

# Generate plots for each CI method using the combined metrics data
wald_plot <- plot_probability(all_metrics_combined, "Wald")
adj_wald_plot <- plot_probability(all_metrics_combined, "Adjusted Wald")
exact_plot <- plot_probability(all_metrics_combined, "Clopper-Pearson")
score_plot <- plot_probability(all_metrics_combined, "Score")
bootstrap_raw_plot <- plot_probability(all_metrics_combined, "Bootstrap Raw")
bootstrap_t_plot <- plot_probability(all_metrics_combined, "Bootstrap t")

# Arrange the generated plots in a grid format with 3 columns
# The grid.arrange function displays multiple plots together
grid.arrange(wald_plot, adj_wald_plot, exact_plot, score_plot, bootstrap_raw_plot,bootstrap_t_plot,  nc
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 2 rows containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 45 rows containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



Results of the simulation

Conclusion

Wald Method

- Coverage

The Wald method shows under-coverage at extreme p values. At p=0.01 and p=0.99, where the probabilities are significantly lower than the target 95%. This indicates that the Wald intervals are too narrow, especially for small n.

- Conclusion

This method is unreliable, especially for small sample sizes and extreme p values. It produces intervals that are too narrow, leading to under-coverage.

Adjusted Wald Method

- Coverage

The Adjusted Wald method generally shows better coverage than the regular Wald method, especially at extreme p values like 0.01 and 0.99. For example, at n=100, p=0.99, the Adjusted Wald method achieves nearly 99% coverage.

- Conclusion

This is an improved version of the Wald method and provides better coverage, especially for extreme p values. However, its intervals are longer than the regular Wald intervals.

Clopper-Pearson (Exact) Method

- Coverage

The Clopper-Pearson method consistently has low coverage at mid-range p values (e.g., p=0.5), indicating that it produces overly conservative (narrow) intervals. Coverage values range from 0.492 to 0.584, significantly lower than 0.95.

- Conclusion

While this method is conservative, it tends to under-cover at mid-range probabilities. Its overly conservative intervals result in poor performance at p=0.5.

Score Method

- Coverage

The Score method achieves better coverage than the Wald method, particularly for moderate p values like 0.255 and 0.745, where it consistently reaches coverage near 0.96 or above. It also handles extreme p values better than Wald.

- Conclusion

The Score method is one of the most reliable methods, providing good coverage and reasonable interval lengths. It performs well across different sample sizes and probabilities.

Bootstrap Raw Percentile Method

- Coverage

The Bootstrap Raw Percentile method mostly fails to provide sufficient coverage, especially for small sample sizes. For n=15 and n=30, it returns zero intervals for many p values, which suggest that it fail to estimate bounds effectively.

- Conclusion

This method is unreliable, especially for smaller sample sizes.

Bootstrap t Method

- Coverage

The Bootstrap t method shows excellent coverage for moderate p values and larger sample sizes (ex., p=0.255/0.5/0.745 with n=100), with coverage reaching 1.0.

- Conclusion

This method performs well for moderate p values and larger sample sizes but tends to produce excessively wide intervals, leading to over-coverage in some cases.

For balanced performance across a variety of sample sizes and p values, the Score method appears to be the most reliable, while the Bootstrap t method may be useful for larger samples if more conservative intervals are desired.