Bootstrap

2024-05-22

Bootstrapping is a powerful statistical technique used to estimate the sampling distribution of a statistic by repeatedly resampling with replacement from the original data. This method allows for the assessment of the variability and accuracy of the statistic without relying on strong parametric assumptions. By generating numerous resampled data points, bootstrapping creates a distribution of the statistic of interest, which can be used to derive confidence intervals and other measures of statistical uncertainty. This technique is particularly useful when the theoretical distribution of the statistic is complex or unknown.

In our analysis, bootstrapping is applied to estimate the efficacy of the BNT162b2 vaccine. By generating multiple resampled datasets from the original data, we can repeatedly calculate the vaccine efficacy and construct a distribution of these efficacy estimates. This distribution provides valuable insights into the range, variability, and confidence intervals of the vaccine's effectiveness. Using bootstrapping, we can assess the reliability of the observed efficacy, ensuring that the conclusions drawn from the data are well-supported by a rigorous statistical framework.

```r
data <- read_csv("data.csv", show_col_types = F)
```

```r
vaccine <- data %>%
  filter(Test == "Vaccine")

placebo <- data %>%
  filter(Test == "Placebo")

prop_vaccine <- vaccine$COVID / (vaccine$COVID + vaccine$No_COVID)
prop_placebo <- placebo$COVID / (placebo$COVID + placebo$No_COVID)

observed_pi <- prop_vaccine/(prop_vaccine + prop_placebo)

observed_psi <- (1 - 2*observed_pi)/(1 - observed_pi)
```

```r
n_bootstrap <- 10000
bootstrap_psis <- numeric(n_bootstrap)
set.seed(123)

for (i in 1:n_bootstrap) {
  vaccine_sample <- sample(c(0, 1), size = vaccine$COVID + vaccine$No_COVID, replace = TRUE, prob = c(1
  placebo_sample <- sample(c(0, 1), size = placebo$COVID + placebo$No_COVID, replace = TRUE, prob = c(1

  prop_vaccine_boot <- mean(vaccine_sample)
  prop_placebo_boot <- mean(placebo_sample)

  bootstrap_pi <- prop_vaccine_boot / (prop_vaccine_boot + prop_placebo_boot)

  bootstrap_psis[i] <- (1 - 2 * bootstrap_pi) / (1 - bootstrap_pi)
}
```

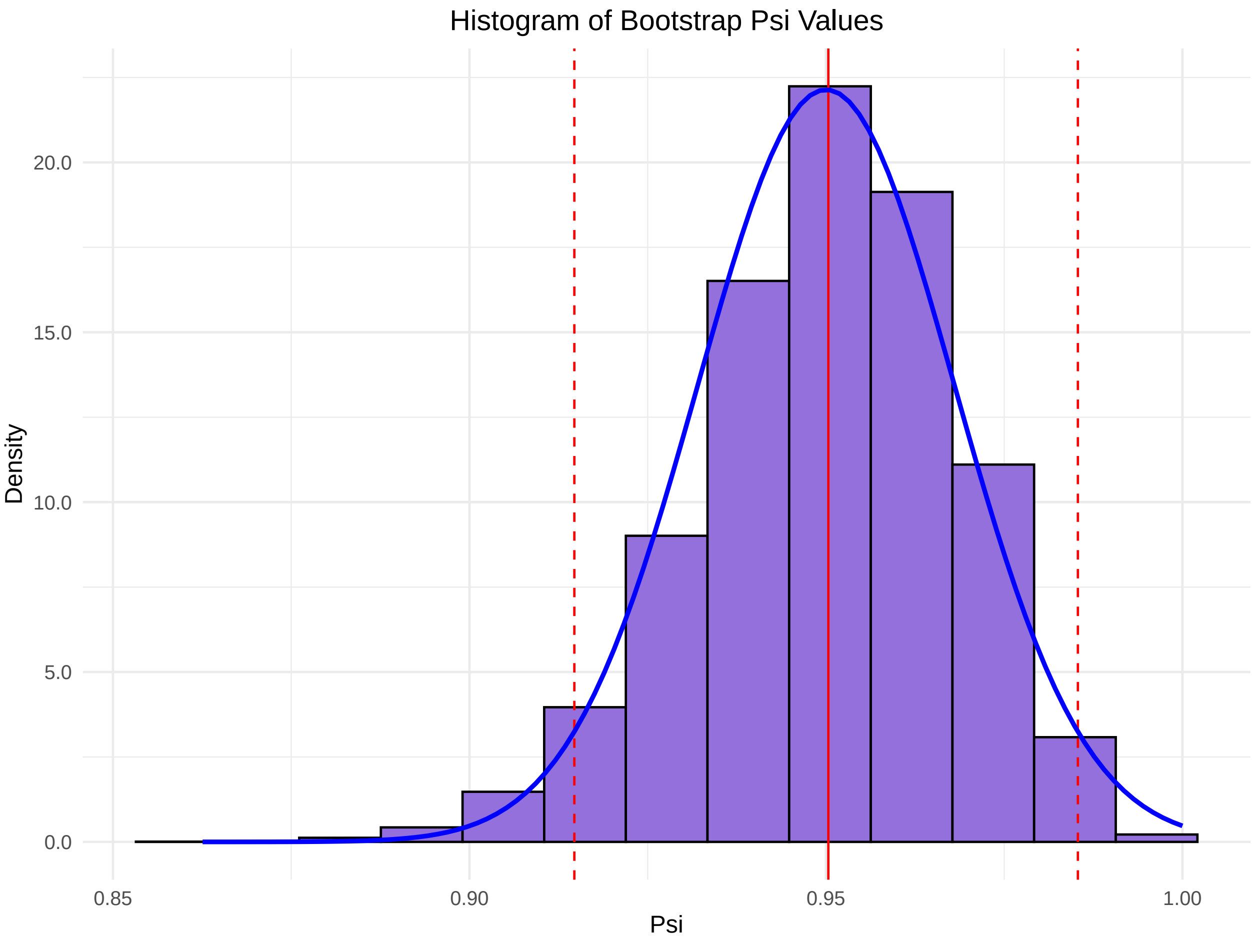
```r
bootstrap_df <- data.frame(psi = bootstrap_psis)
```

```
bootstrap_summary <- bootstrap_df %>%
  summarise(n = n(),
            mean = mean(psi),
            sd = sd(psi),
            lower_critical_value = mean - qnorm(0.975)*sd,
            upper_critical_value =  mean + qnorm(0.975)*sd)

lower_critical_value <- bootstrap_summary$lower_critical_value
upper_critical_value <- bootstrap_summary$upper_critical_value

bootstrap_summary
```

```
##       n      mean         sd lower_critical_value upper_critical_value
## 1 10000 0.9500209 0.01801763             0.914707            0.9853348
```

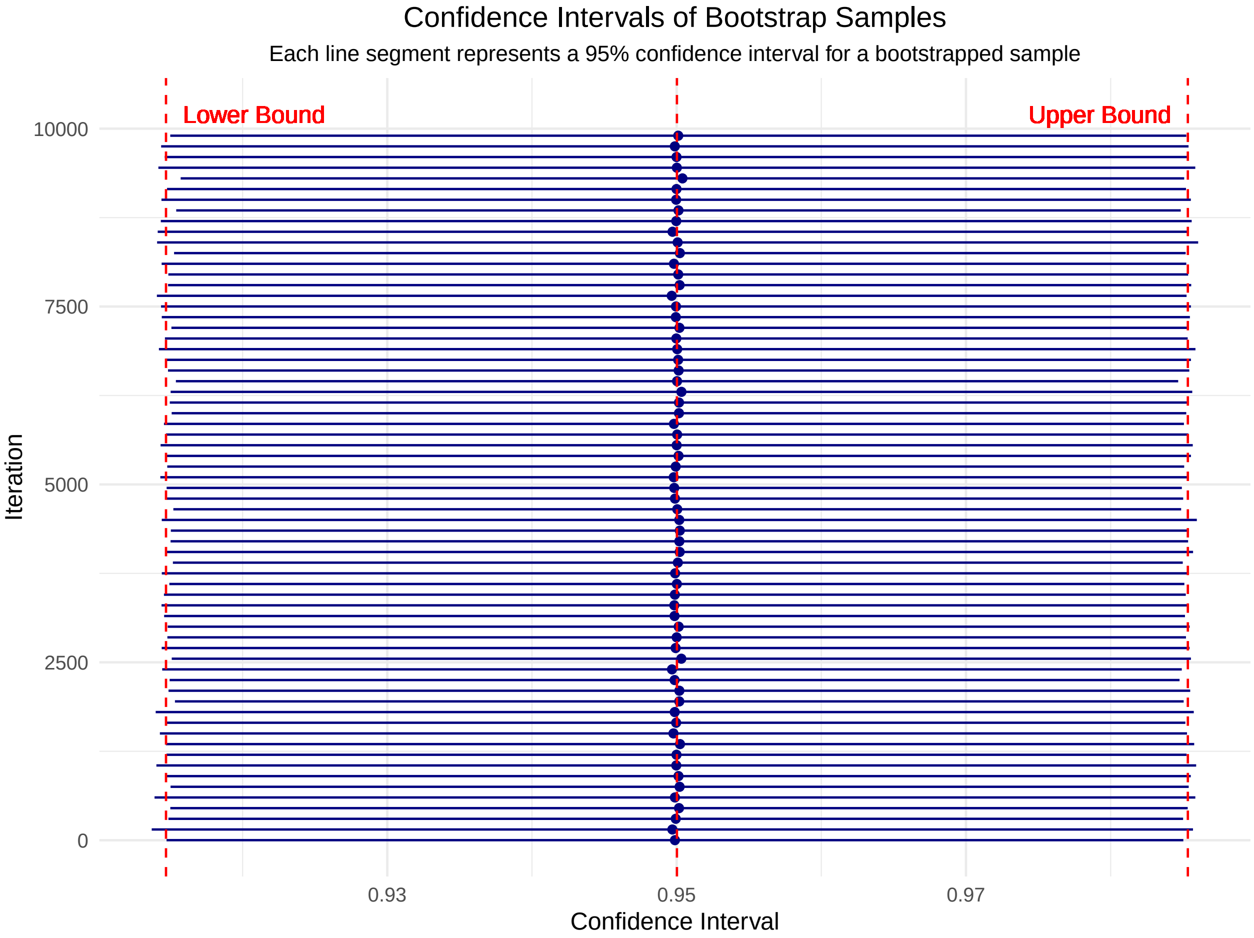
```
ggplot(bootstrap_df, aes(x = psi)) +
  geom_histogram(aes(y = after_stat(density)),
                 bins = round(log(length(bootstrap_df$psi), base = 2)),
                 fill = "mediumpurple",
                 color = "black") +
  labs(title = "Histogram of Bootstrap Psi Values", x = "Psi", y = "Density") +
  theme_minimal() +
  geom_vline(xintercept = observed_psi,
             linetype = "solid",
             color = "red") +
  geom_vline(xintercept = bootstrap_summary$upper_critical_value,
             linetype = "dashed",
             color = "red") +
  geom_vline(xintercept = bootstrap_summary$lower_critical_value,
             linetype = "dashed",
             color = "red") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.1)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  stat_function(fun = dnorm, args = list(mean = bootstrap_summary$mean, sd = bootstrap_summary$sd), colo
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```r
ci_data <- data.frame(
  Iteration = 1:n_bootstrap,
  Lower = numeric(n_bootstrap),
  Upper = numeric(n_bootstrap)
)

for (i in 1:n_bootstrap) {
  sample_psis <- sample(bootstrap_psis, n_bootstrap, replace = TRUE)
  ci_data$Lower[i] <- mean(sample_psis) - qnorm(0.975)*sd(sample_psis)
  ci_data$Upper[i] <- mean(sample_psis) + qnorm(0.975)*sd(sample_psis)
  ci_data$Mean[i] <- mean(sample_psis)
}
```

```r
plot_data <- ci_data[seq(1, n_bootstrap, by = 150), ]

ggplot(plot_data, aes(y = Iteration)) +
  geom_segment(aes(yend = Iteration, x = Lower, xend = Upper), color = "navy") +
  geom_point(aes(y = Iteration, x = Mean), color = "navy")+
  geom_vline(xintercept = lower_critical_value, linetype = "dashed", color = "red") +
  geom_vline(xintercept = upper_critical_value, linetype = "dashed", color = "red") +
  geom_text(aes(x = lower_critical_value + 0.012, y = max(Iteration) + 300, label = "Lower Bound"), colo
  geom_text(aes(x = upper_critical_value - 0.012, y = max(Iteration) + 300, label = "Upper Bound"), colo
  geom_vline(xintercept = bootstrap_summary$mean, linetype = "dashed", color = "red") +
  labs(title = "Confidence Intervals of Bootstrap Samples",
       y = "Iteration",
       x = "Confidence Interval",
       subtitle = "Each line segment represents a 95% confidence interval for a bootstrapped sample") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5, size = 10),
        axis.text.y = element_text(hjust = 1))
```



```
p_val <- pnorm(0.3, mean = bootstrap_summary$mean, sd = bootstrap_summary$sd, lower.tail = T)

emp_p_val <- length(bootstrap_df[bootstrap_df$psi <= 0.3])/length(bootstrap_df$psi)

cat("Theoretical p-value:", p_val, "\n")
```

```
## Theoretical p-value: 2.607415e-285
```

```
cat("Empirical p-value:", emp_p_val, "\n")
```

```
## Empirical p-value: 0
```