Placebo-Controlled Efficacy Analysis of the COVID-19 Vaccine

Spring 2024

Oliver Brown, Josie Czeskleba, Luke VanHouten

Abstract

Keywords

Efficacy, Inference, COVID-19, Statistics, Estimators

Introduction

In this project, we will be analyzing the efficacy of the Pfizer-BioNTech BNT162b2 COVID-19 mRNA vaccine based on a sample of placebo-controlled COVID tests (Polack et al. 2020). This data was collected in late 2020 among individuals at least sixteen years old who received two doses of the vaccine three weeks apart, with them being tested for COVID afterwards. The efficacy of the vaccine is extremely important, because the goal of vaccination efforts are to save lives. So, we perform statistical testing to identify whether or not the efficacy rate shown in the vaccine data is acceptable. Our hypothesis is that the BNT162B2 vaccine efficacy is 95%, which we will test using different statistical methods.

Statistical Methods

We denote the random variable T as the number of vaccinated individuals from the 170 COVID cases.

$$T \sim Binom(n = 170, \pi)$$

We can define $\pi = \text{P(Vaccine|COVID)} = \frac{\pi_1}{\pi_1+\pi_2}$, given that the sample sizes for the vaccine and placebo groups are approximately equal. Here, π_1 is the proportion of vaccinated individuals who got COVID and π_2 is the proportion of unvaccinated individuals who got COVID. Moreover, we define the vaccine efficacy as $\psi = \frac{1-2\pi}{1-\pi}$ (Senn 2021). We can then formulate the following hypothesis test:

$$H_0 : \psi_0 = 0.95$$
$$H_1 : \psi_0 \neq 0.95$$

Maximum Likelihood Estimator

We can first write the likelihood function of π

$$L(\pi) = \binom{n}{t}\pi^t(1-\pi)^{n-t}$$

Then we write π in the form $\pi = g(\psi)$, given that $\psi = \frac{1-2\pi}{1-\pi}$. We thus have that $\psi - \psi\pi = 1 - 2\pi$, which becomes $2\pi - \psi\pi = 1 - \psi$, which becomes:

$$\pi = \frac{1-\psi}{2-\psi}$$

We can then write the likelihood function for ψ:

$$L(\psi) = L(g(\psi)) = L\left(\frac{1-\psi}{2-\psi}\right) = \binom{n}{t}\left(\frac{1-\psi}{2-\psi}\right)^t\left(1-\left(\frac{1-\psi}{2-\psi}\right)\right)^{n-t} = \binom{n}{t}\left(\frac{1-\psi}{2-\psi}\right)^t\left(\frac{1}{2-\psi}\right)^{n-t}$$

We can then calculate the log-likelihood function for ψ:

$$\ell(\psi) = \ln(L(\psi) = \ln\left(\binom{n}{t}\right) + t\ln(1-\psi) - t\ln(2-\psi) - (n-t)\ln(2-\psi) = \ln\left(\binom{n}{t}\right) + t\ln(1-\psi) - n\ln(2-\psi)$$

We can then find our estimator by setting $\ell'(\psi) = 0$:

$$\frac{d}{d\psi}\ell(\psi) = \frac{d}{d\psi}\ln\left(\binom{n}{t}\right) + \frac{d}{d\psi}t\ln(1-\psi) - \frac{d}{d\psi}n\ln(2-\psi) = \frac{n}{2-\psi} - \frac{t}{1-\psi} = 0$$

We can then solve $\frac{n}{2-\psi} = \frac{t}{1-\psi}$. We get that $n - n\psi = 2t - t\psi$, which becomes $t\psi - n\psi = 2t - n$, giving us an estimator of $\widehat{\psi}_0^{mle} = \frac{2t-n}{t-n}$. We can use our MLE to perform a likelihood ratio test, defined as $W = 2\left(\ell\left(\hat{\psi}_0^{mle}\right) - \ell\left(\psi_0^{null}\right)\right)$, which we can write as:

$$W = 2\left(\left(\ln\left(\binom{n}{t}\right) + t\ln\left(1-\widehat{\psi}_0^{mle}\right) - n\ln\left(2-\widehat{\psi}_0^{mle}\right)\right) - \left(\ln\left(\binom{n}{t}\right) + t\ln\left(1-\psi_0^{null}\right) - n\ln\left(2-\psi_0^{null}\right)\right)\right)$$

This becomes:

$$W = 2t\ln\left(1-\widehat{\psi}_0^{mle}\right) - 2n\ln\left(2-\widehat{\psi}_0^{mle}\right) - 2t\ln\left(1-\psi_0^{null}\right) + 2n\ln\left(2-\psi_0^{null}\right)$$

Bootstrap

The bootstrap approach involves repeatedly resampling the observed data with replacement to create numerous simulated datasets. For each simulated dataset, we compute the proportions of COVID-19 cases in both groups, and subsequently, the efficacy parameter ψ. This process provides a distribution of the efficacy estimates from which we can derive confidence intervals.

Step-wise we begin by creating two subsets of the data: one for the vaccine group and one for the placebo group. This step ensures that we can accurately calculate the number of subjects and the proportions of COVID-19 cases within each group.

Once the proportions are calculated, we compute the observed efficacy parameter π and subsequently ψ. The parameter π is defined as the proportion of COVID-19 cases in the vaccine group divided by the sum of the proportions in both the vaccine and placebo groups. The efficacy parameter ψ is then calculated using the formula $\psi = \frac{1-2\pi}{1-\pi}$. These parameters provide a basis for comparing the vaccine's efficacy against COVID-19.

To assess the variability of the efficacy estimate, we perform a bootstrap simulation with 10,000 iterations. In each iteration, resample the data with replacement to generate new datasets for both the vaccine and placebo groups. For each bootstrap sample, we recalculate the proportions of COVID-19 cases and subsequently the efficacy parameter ψ. This process generates a distribution of ψ values, which can be used to estimate the confidence interval.

We calculate the 95% confidence interval for the efficacy parameter ψ using the quantiles of the bootstrap distribution. This interval provides a range within which the true efficacy is likely to lie, based on the variability observed in the bootstrap samples.

Results

For our MLE, we can plug in $t_{obs} = 8$ and $n = 170$ to $\widehat{\psi}_0^{mle} = \frac{2t-n}{t-n}$, we get $\widehat{\psi}_0^{mle} = \frac{16-170}{8-170} = \frac{77}{81} = 0.9506$. We can also use the Newton Raphson method to estimate ψ to get the same value, shown in the appendix.

For the likelihood ratio test, we can plug in our values $\widehat{\psi}_0^{mle} = 0.9506$, $\psi_0^{null} = 0.95$, $t = 8$, and $n = 170$ to our W to get:

$$W = 2(8)\ln(1 - 0.9506) - 2(170)\ln(2 - 0.9506) - 2(8)\ln(1 - 0.95) + 2(170)\ln(2 - 0.95) = 0.0012$$

Our sample size is large enough to where $W \sim \chi_1^2$, so we can calculate P-value for our hypothesis as $P(W \geq 0.0012) = 0.9726$. This P-value is very large, so we fail to reject the null hypothesis under our likelihood ratio test that the COVID-19 vaccine efficacy is 95%.

For our bootstrap, the analysis was performed with 10000 iterations to ensure a stable estimate of the vaccine efficacy. The observed efficacy, calculated from the original data, was consistent with the findings of Polack et al. (2020). The histogram of the bootstrap ψ values revealed a right-skewed distribution, indicating that most of the bootstrap samples support a high efficacy of the vaccine. The 95% confidence interval for ψ derived from the bootstrap distribution, ranged from 0.91 to 0.98, closely aligning with the Bayesian credible interval reported in the original study.

(Histogram)

(Segmented line graph)

Conclusion

References

Lee, Jack C., and Darius J. Sabavala. 1987. "Bayesian Estimation and Prediction for the Beta-Binomial Model." *Journal of Business & Economic Statistics* 5 (3): 357–67. http://www.jstor.org/stable/1391611.

Polack, Fernando P., Stephen J. Thomas, Nicholas Kitchin, Judith Absalon, Alejandra Gurtman, Stephen Lockhart, John L. Perez, et al. 2020. "Safety and Efficacy of the BNT162b2 mRNA Covid-19 Vaccine." *New England Journal of Medicine* 383 (27): 2603–15. https://doi.org/10.1056/NEJMoa2034577.

Senn, Stephen. 2021. "S. Senn: 'Beta Testing': The Pfizer/BioNTech Statistical Analysis of Their Covid-19 Vaccine Trial (Guest Post)." *Error Statistics Philosophy*. https://errorstatistics.com/2021/01/17/s-senn-beta-testing-the-pfizer-biontech-statistical-analysis-of-their-covid-19-vaccine-trial-guest-post/.

Appendix

Newton Rhapson MLE Approximation

```
loglik <- function(psi, T, n){
  if (psi > 1 | psi < 0)
    return(NA)
  else
    return(log(choose(n, T)) + (T * log(1 - psi)) - (n * log(2 - psi)))
}

estimation <- maxLik2(loglik = loglik, start = 0.55, method = "NR", tol = 1e-4,
                      T = 8, n = 170)

print(estimation)
```

```
## Maximum Likelihood estimation
## Newton-Raphson maximisation, 6 iterations
## Return code 2: successive function values within tolerance limit (tol)
## Log-Likelihood: -1.944994 (1 free parameter(s))
## Estimate(s): 0.9506174
```

Likelihood Ratio Test Calculation

```
W = (2 * 8 * log(1 - (77 / 81))) - (2 * 170 * log(2 - (77 / 81))) -
    (2 * 8 * log(1 - 0.95)) + (2 * 170 * log(2 - 0.95))

p_value <- round(pchisq(q = W, df = 1, lower.tail=F), 4)
```

Here, W is 0.0012 and the corresponding P-value is 0.9726.

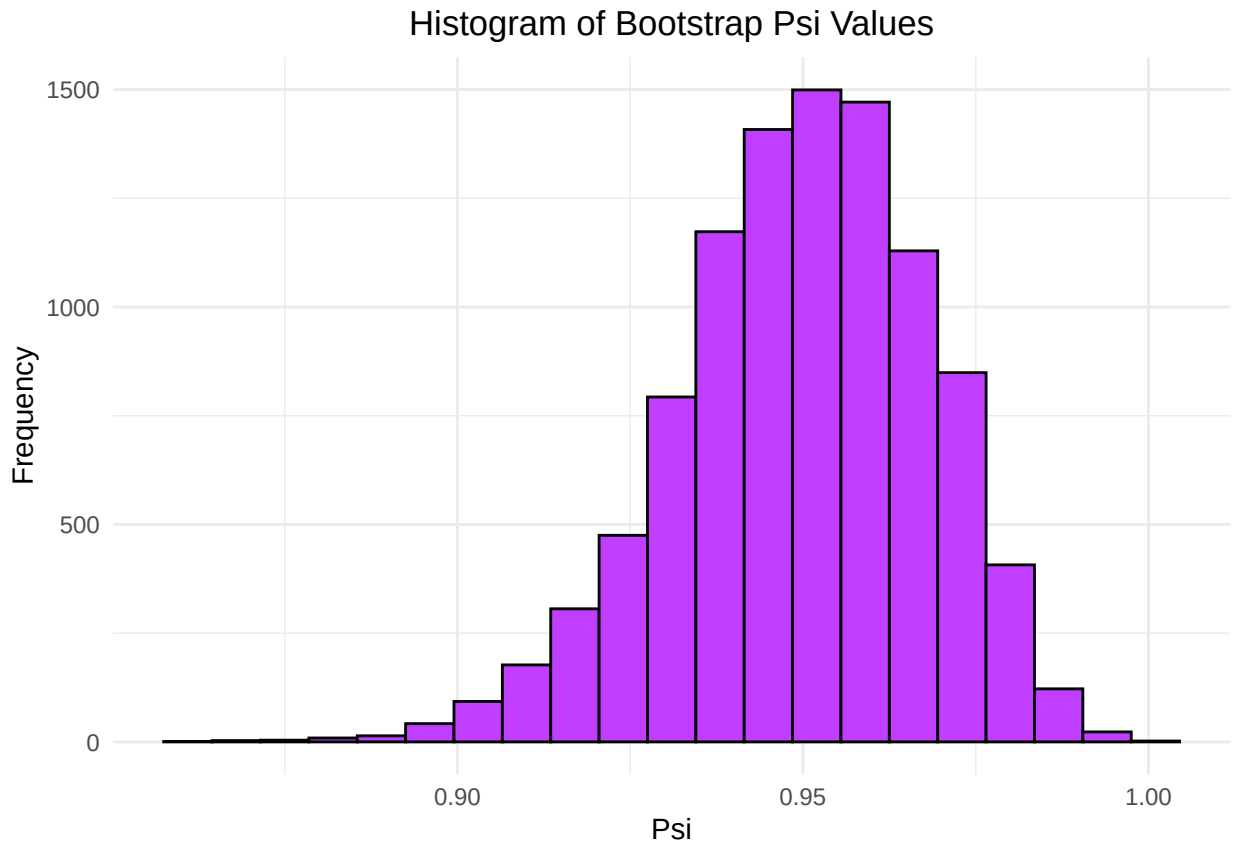
Bootstrap

```
data <- read.csv("data.csv")

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

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

n_vaccine <- vaccine$COVID + vaccine$No_COVID
n_placebo <- placebo$COVID + placebo$No_COVID

prop_vaccine <- vaccine$COVID[1] / n_vaccine
prop_placebo <- placebo$COVID[1] / n_placebo

observed_pi <- prop_vaccine/(prop_vaccine + prop_placebo)

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

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

for (i in 1:n_bootstrap) {
  vaccine_sample <- sample(c(0, 1), size = n_vaccine, replace = TRUE,
                           prob = c(1 - prop_vaccine, prop_vaccine))
  placebo_sample <- sample(c(0, 1), size = n_placebo, replace = TRUE,
                           prob = c(1 - prop_placebo, prop_placebo))

  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)
}
```

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

ggplot(bootstrap_df, aes(x = psi)) +
  geom_histogram(binwidth = 0.007, fill = "darkorchid1", color = "black") +
  labs(title = "Histogram of Bootstrap Psi Values", x = "Psi", y = "Frequency") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

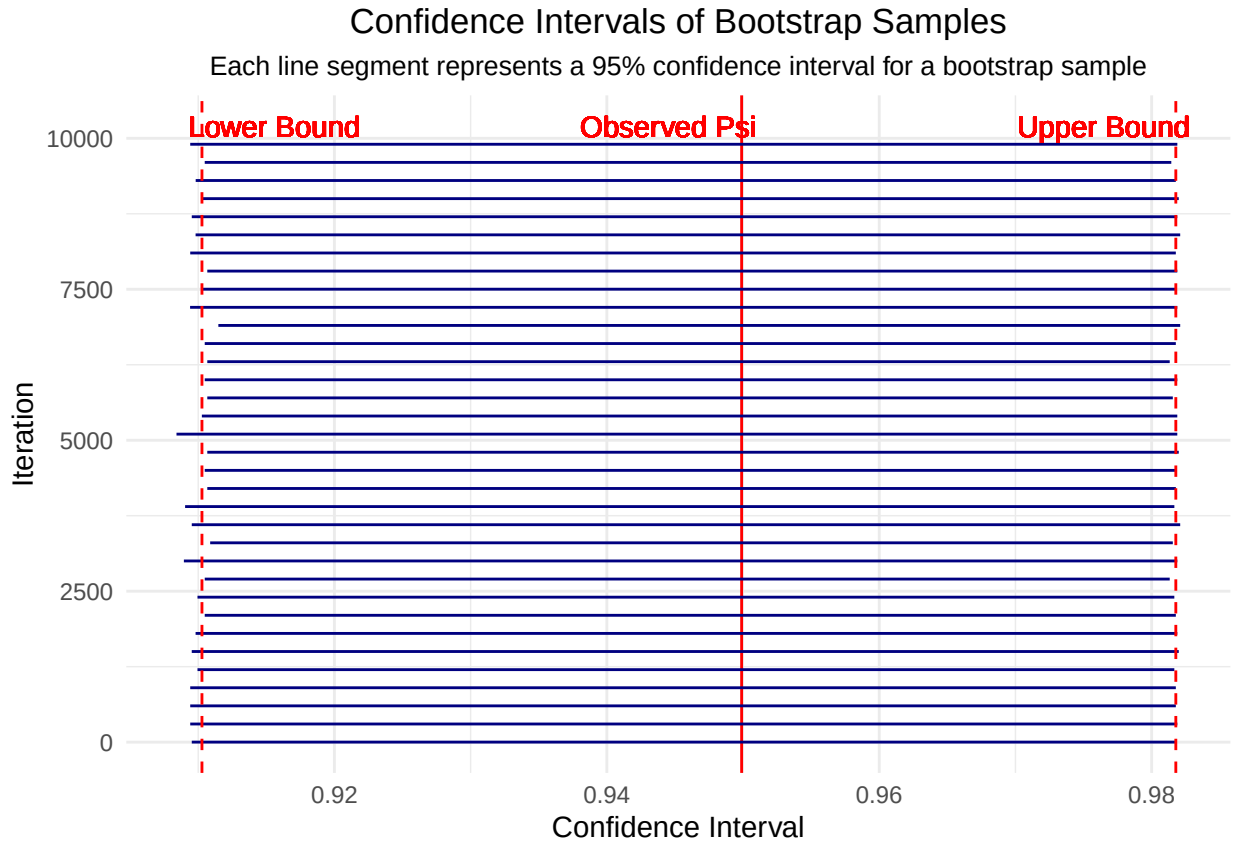



```
overall_ci <- quantile(bootstrap_psis, c(0.025, 0.975))

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] <- quantile(sample_psis, 0.025)
```

```r
  ci_data$Upper[i] <- quantile(sample_psis, 0.975)
}

print(overall_ci)
```

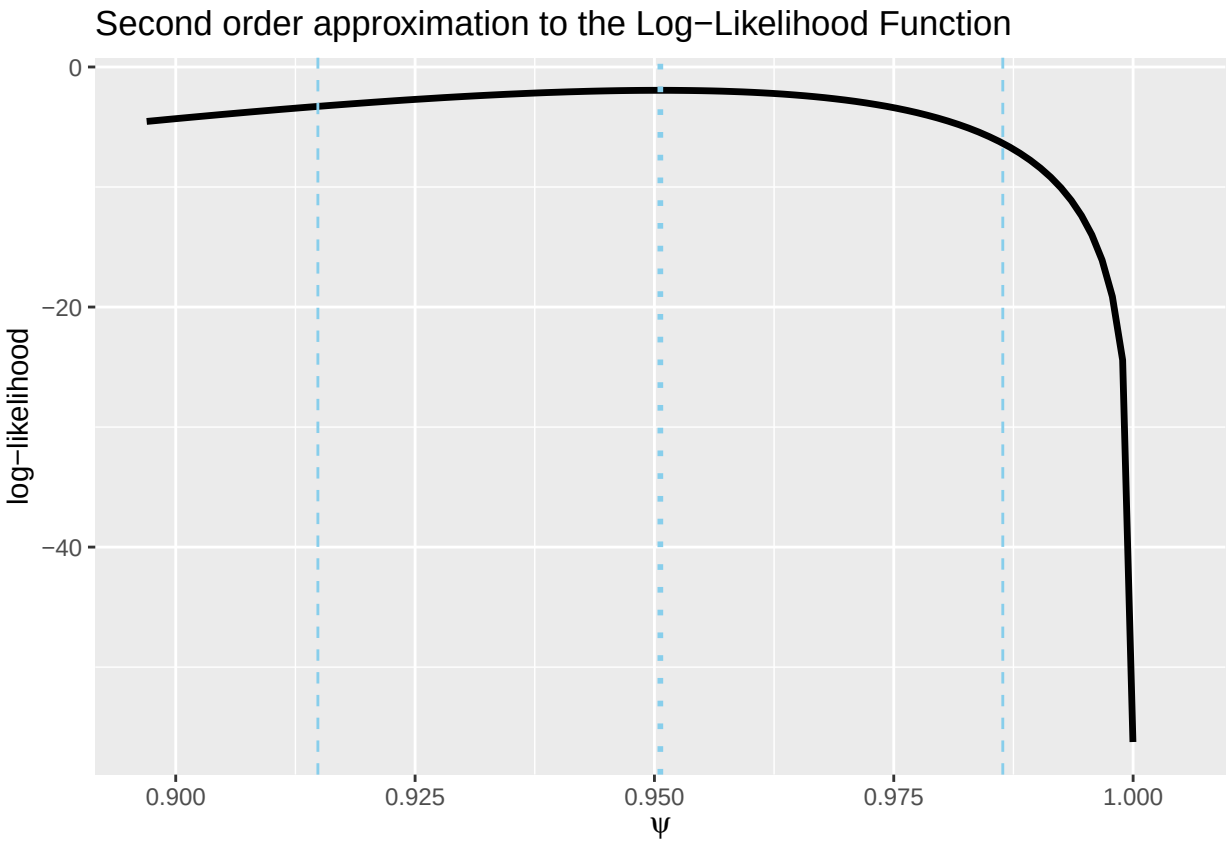
```
##      2.5%     97.5%
## 0.9102751 0.9817740
```

```r
plot_data <- ci_data[seq(1, n_bootstrap, by = 300), ]

ggplot(plot_data, aes(y = Iteration)) +
  geom_vline(xintercept = observed_psi, linetype = "solid", color = "red") +
  geom_text(aes(x = observed_psi-0.0125,
                y = max(Iteration) + 300, label = "Observed Psi"),
            color = "red", hjust = -0.05) +
  geom_segment(aes(yend = Iteration, x = Lower, xend = Upper), color = "navy") +
  geom_vline(xintercept = overall_ci[1], linetype = "dashed", color = "red") +
  geom_vline(xintercept = overall_ci[2], linetype = "dashed", color = "red") +
  geom_text(aes(x = overall_ci[1] + 0.0129,
                y = max(Iteration) + 300, label = "Lower Bound"),
            color = "red", hjust = 1.1) +
  geom_text(aes(x = overall_ci[2] - 0.0129, y = max(Iteration) + 300,
                label = "Upper Bound"), color = "red", hjust = -0.1) +
  labs(title = "Confidence Intervals of Bootstrap Samples",
       y = "Iteration",
       x = "Confidence Interval",
       subtitle = "Each line segment represents a 95% confidence interval for a bootstrap sampl
  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))
```



Visualizations

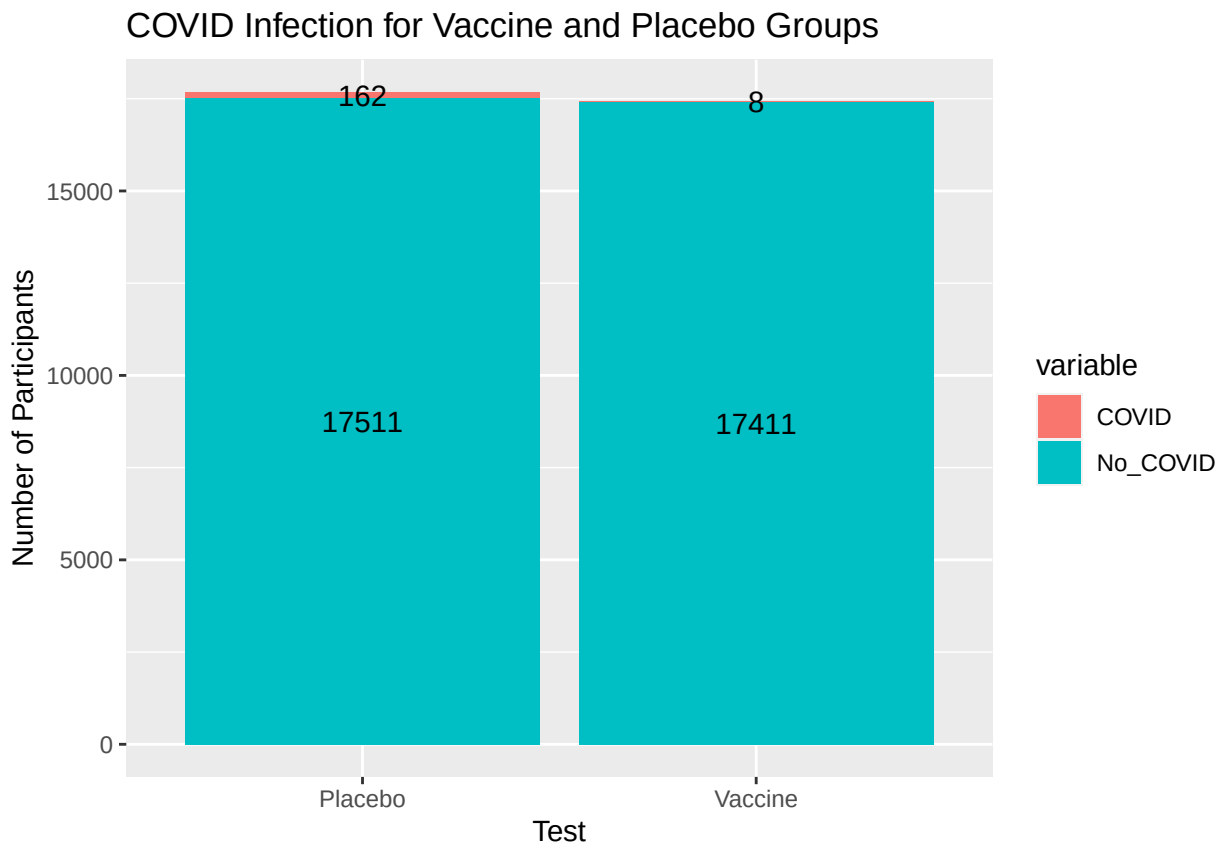
```
plot(estimation) +
    labs(title = "Second order approximation to the Log-Likelihood Function",
         x = expression(psi))
```




```
data_melted <- melt(data, id.vars = "Test")
```

Stacked Barplot

```
ggplot(data_melted, aes(x = Test, y = value, fill = variable)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = value),
            position = position_stack(vjust = 0.5)) +
  labs(x = "Test", y = "Number of Participants",
       title = "COVID Infection for Vaccine and Placebo Groups")
```



Box Plot

```
pivot_data <- data %>%
  pivot_longer(cols = c(COVID, No_COVID), names_to = "Tests", values_to = "Count")

summary <- pivot_data %>%
    group_by(Test) %>%
    summarise(min = min(Count), lower = quantile(Count, 0.25),
              median = median(Count), upper = quantile(Count, 0.75),
              max = max(Count))

ggplot(pivot_data, aes(x = Test, y = Count, fill = Test)) +
    geom_boxplot() +
    geom_text(data = summary, aes(x = Test, y = median,
                                  label = paste("Median:", median)),
              vjust = -0.5) +
    geom_text(data = summary, aes(x = Test, y = lower,
                                  label = paste("Q1:", lower)),
              vjust = 1.5, hjust = -0.3) +
    geom_text(data = summary, aes(x = Test, y = upper,
                                  label = paste("Q3:", upper)),
```

```
            vjust = -1, hjust = -0.3) +
  geom_text(data = summary, aes(x = Test, y = min,
                                label = paste("Min:", min)),
            vjust = 1.3) +
  geom_text(data = summary, aes(x = Test, y = max,
                                label = paste("Max:", max)),
            vjust = -0.5) +
  labs(title = "Positive and Negative COVID Test Counts by Test Group",
       x = "Test Group", y = "Number of Tests")
```