Title

Alexandra Orczyk, Vy Nguyen, Xiyue Huang

1 Abstract:

2 Keywords:

3 Introduction:

4 Statistical Methods:

Method 2: Method of Moment Inference for Vaccine efficacy

We are interested in the value of $\hat{\phi}_0^{mom}$. Since $\pi = \frac{1-\psi}{2-\psi}$, we can find the M.O.M estimate of ψ_0 by solving the equation $E(X) = \bar{x}$, where $\bar{x}$ equals to x in our data.

$$\begin{aligned} E(X) &= x \\ n\pi &= x \\ n\frac{1-\psi}{2-\psi} &= x \\ 2x - x\psi &= n - n\psi \\ \psi(n-x) &= n - 2x \\ \psi &= \frac{n-2x}{n-x} \end{aligned}$$

Therefore, $\hat{\psi}_0^{mom} = \frac{n-2x}{n-x}$. Now, in order to make inferences about ψ(the vaccine efficacy) using the method of moments (MOM) approach, we can construct confidence intervals (CI) and conduct hypothesis testing.

To construct confidence intervals for ψ, we will use bootstrapping method. In the context of vaccine efficacy, we will first randomly re-sample the original data with replacement. Second, for each re-sample, calculate the desired statistic (e.g.,ϕ, the vaccine efficacy). Then, repeat the re-sampling process many times (e.g., 10,000 times) to build a distribution of the statistic. Finally, using the bootstrap distribution to calculate confidence intervals of ϕ:

$$\hat{\phi}_0^{mom} \pm Z_{\alpha/2} SE(\hat{\phi}_0^{mom})$$

where α is the significant level.

Another statistical inference of our estimation is hypothesis testing. In the context of vaccine efficacy, let ϕ_0 denote the true (but unknown) value of vaccine efficacy. We want to test:

$H_0 : \phi_0 = \phi_0^{null}$,

$H_1 : \phi_0 \neq \phi_0^{null}$

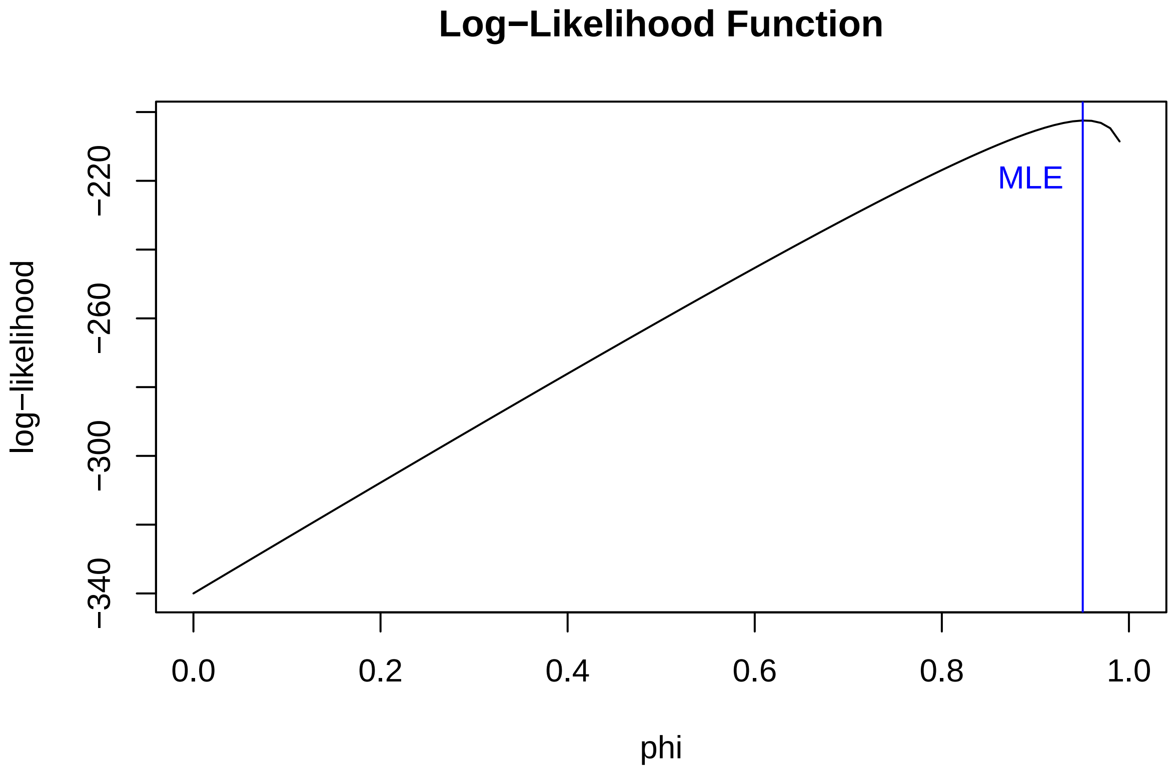
By using the empirical p-value from the bootstrap distribution, we can make a more robust inference about the vaccine efficacy ϕ. If the empirical p-value is less than the significance level, we reject the null hypothesis and conclude that the vaccine has a significant effect.

5 Resutls:

Plot for log likelihood function

```
# Define the log-likelihood function
loglik.binom <- function(phi, x, n) {
  ifelse((phi < 0 | phi > 1), NA, x * log(1 - phi) - n * (2 - phi))
}

# Example data
n <- 170
```

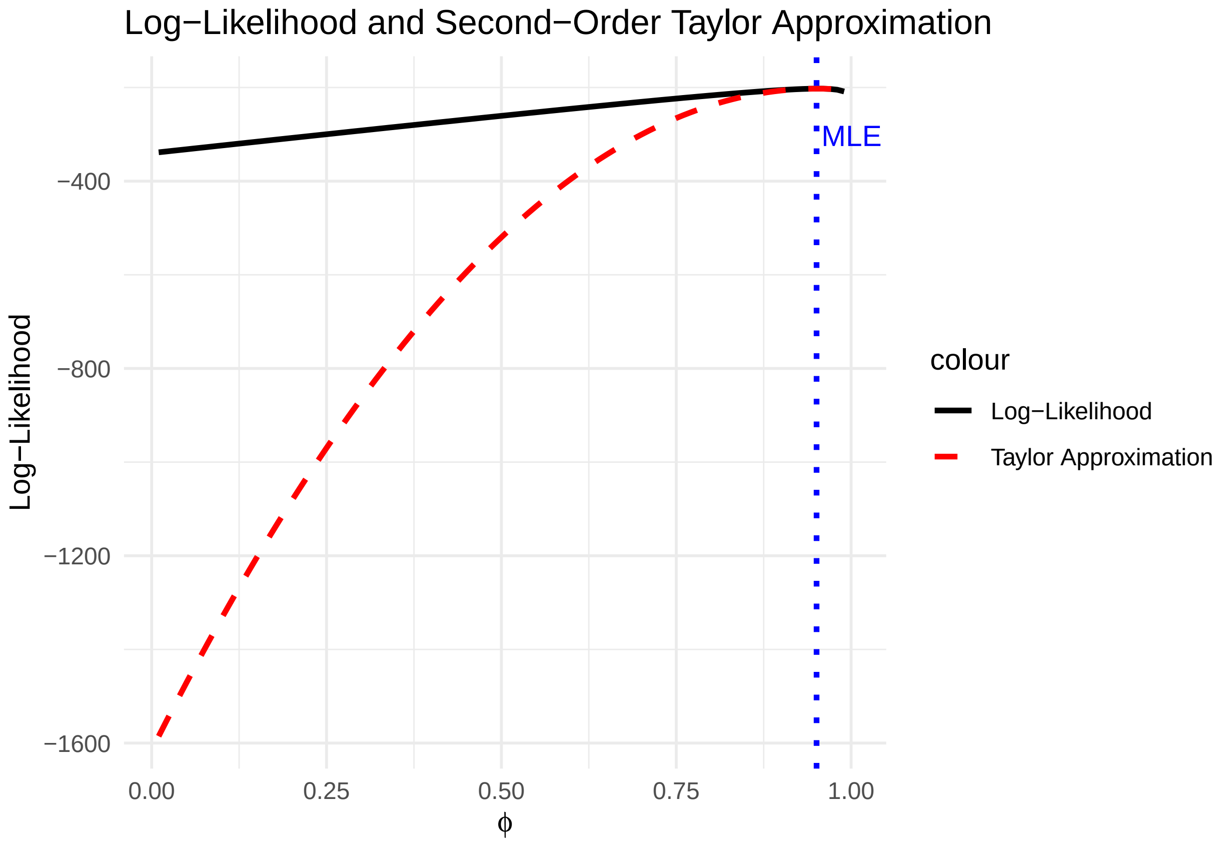
```
x <- 8


theta_0_mle_estimate <- (n-2*x)/(n-x)


phi_values <- seq(0, 1, by = 0.01)
log_likelihood_values <- sapply(phi_values, function(phi) loglik.binom(phi, x, n))


# plot of log likelihood function for phi
plot(phi_values, log_likelihood_values, type = "l",
     xlab = "phi", ylab = "log-likelihood", main = "Log-Likelihood Function")
abline(v = theta_0_mle_estimate, col = "blue")
text(x = theta_0_mle_estimate, y = max(-340:-220), labels = "MLE", pos = 2, col = "blue")
```



Regularity Condition Check

```r
# Define the second-order Taylor approximation function
taylor_approx <- function(phi, x, n, phi_hat) {
  loglik_prime <- x / (1 - phi_hat) - n / (2 - phi_hat)
  loglik_double_prime <- -x / (1 - phi_hat)^2 + n / (2 - phi_hat)^2
  return(loglik.binom(phi_hat, x, n) + loglik_prime * (phi - phi_hat) + 0.5 * loglik_double_pri
}

phi_values <- seq(0.01, 0.99, by = 0.01)

# Compute the log-likelihood values and the second-order Taylor approximation
loglik_values <- loglik.binom(phi_values, 8, 170)
taylor_values <- taylor_approx(phi_values, 8, 170, theta_0_mle_estimate)

# Create a dataframe for plotting
df <- data.frame(phi = phi_values, loglik = loglik_values, taylor = taylor_values)

# Plot
ggplot(df, aes(phi)) +
  geom_line(aes(y = loglik, color = "Log-Likelihood"), size = 1) +
  geom_line(aes(y = taylor, color = "Taylor Approximation"), linetype = "dashed", size = 1) +
  geom_vline(xintercept = theta_0_mle_estimate, color = "blue", linetype = "dotted", size = 1)
  labs(title = "Log-Likelihood and Second-Order Taylor Approximation",
       x = expression(phi),
       y = "Log-Likelihood") +
  scale_color_manual(values = c("black", "red"),
                     labels = c("Log-Likelihood", "Taylor Approximation")) +
  theme_minimal() +
  annotate("text", x = theta_0_mle_estimate + 0.05, y = max(df$loglik, na.rm = TRUE) - 100,
           label = "MLE", color = "blue")
```

```
## 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.
```



MOM

```
# Given data
n_total <- 170
x_obs <- 8
n <- 170

# MOM estimate of vaccine efficacy
psi_mom_obs <- (n - 2 * x_obs) / (n - x_obs)
```

```r
# Number of bootstrap samples
num_bootstraps <- 10000

# Function to calculate psi_mom
calculate_psi_mom <- function(x) {
  return((n - 2 * x) / (n - x))
}

# Generate bootstrap samples and calculate psi_mom for each
bootstrapped_psi_mom <- replicate(num_bootstraps, {
  x_boot <- rbinom(1, n_total, x_obs / n_total)
  calculate_psi_mom(x_boot)
})

# Calculate confidence intervals
lower_bound <- quantile(bootstrapped_psi_mom, 0.025)
upper_bound <- quantile(bootstrapped_psi_mom, 0.975)

# Calculate empirical p-value
p_value <- mean(bootstrapped_psi_mom <= psi_mom_obs)

# Results
c(psi_mom_obs, lower_bound, upper_bound, p_value)
```

```
##                2.5%     97.5%
## 0.9506173 0.9102564 0.9820359 0.5578000
```

The plot will show the 95% confidence interval for the vaccine efficacy estimate, with the observed MOM estimate indicated by a red dashed line and a blue point. This visualization helps to see the variability and central tendency of the MOM estimates obtained from bootstrapping

6 Discussion/Conclusion:

7 References:

8 Appendix: