Re-examining Bangladesh Voter Support: A Bayesian Approach with Bootstrap

Shafqat Shafiq

2025-03-11

Recent discussions have centered on a poll claiming BNP holds 42% voter support, with a counter-analysis asserting the actual support is closer to 17%. The initial poll derived its 42% figure from a subset of 6,631 respondents who had both decided on their voting preference and explicitly stated it. The counter-analysis, however, argues that this is misleading and that the full respondent pool of 10,696 should be used, resulting in the 17% figure.

This paper presents a re-evaluation of the data, focusing on the 6,631 respondents who actively participated and expressed their voting intentions. We contend that using this subset is more appropriate for accurately representing the preferences of those who engaged with the survey. By focusing on this group, we aim to minimize the impact of non-response bias, which occurs when those who do not respond differ systematically from those who do. We will demonstrate why including the entire 10,696 introduces significant distortions, as it assumes that non-respondents share the same voting preferences as respondents—an assumption that is not supported by the data.

Employing a Bayesian statistical framework, we will update our beliefs about BNP's support based on the observed responses of the 6,631 individuals. This methodology involves establishing a prior belief, simulating potential variations in the data through bootstrapping, and calculating a posterior distribution that reflects the observed data. We will also provide a detailed mathematical derivation of the Beta-Binomial posterior, illustrating how our updated beliefs were calculated.

Furthermore, we will address the limitations of using resampling techniques, such as the Bayesian bootstrap, to correct for non-response bias. While bootstrapping is valuable for estimating uncertainty within a given sample, it cannot compensate for the fundamental issue of missing data. Therefore, our focus on the 6,631 respondents is essential for producing a more accurate and defensible analysis. Ultimately, we aim to provide a more nuanced and insightful interpretation of the poll results, highlighting the importance of considering the actively engaged respondents and the limitations of assuming non-response equates to a specific voting preference.

Code:

```
library(tidyverse)
# library(tidymodels)
# library(poissonreg)
# library(rsample)
library(broom)
library(doParallel)
library(boot)
library(pscl)

theme_set(theme_minimal())

# Set default color for points and lines
update_geom_defaults("point", list(color = "midnightblue"))
update_geom_defaults("line", list(color = "midnightblue"))

# Set default fill color for rectangles
update_geom_defaults("rect", list(fill = "midnightblue"))
```

The Beta distribution is used to represent our initial belief about BNP's support. Alpha and beta are parameters of the Beta distribution. Our prior belief about BNP's support is `Beta(alpha_prior, beta_prior)`.

This represents our initial belief based solely on the original survey data, before any bootstrapping.

Step 1: Setting the Prior (Belief before observing new data):

```
set.seed(123)
#---Step 1: Setting the Prior (Belief before observing new data)---

# Original survey data:
total_surveyed <- 10696 # Total number of people surveyed in original poll

total_respondents <- 6631 # Total number of people who actually responded

bnp_supporters <-
  round(total_respondents*.42)
```

```r
# Total number of people who responded to supporting BNP

bnp_supporters
```

```
[1] 2785
```

```r
alpha_prior <-
  bnp_supporters # Alpha represents no. of BNP supporters in original survey
beta_prior <-
  total_respondents - bnp_supporters # Beta represents the non-BNP supporters
```

Step 2: Using Bootstrap to Simulate New Observations:

We will generate 1000 simulated surveys from the original dataset of 6631 respondents and count the number of BNP voters in each bootstrap resample.

```r
# Number of bootstrap
n_bootstrap_samples <- 10000

# Initialize a vector to store the number of BNP supporters in each boot-sample
bnp_counts_bootstrap <-
  numeric(n_bootstrap_samples) # This vector will hold the BNP supporter counts.
```

We now initialize 2 for-loops. The first for-loop will iterate from `1:n_bootstrap_samples` and generate 10,000 pseudo-random samples. The second for-loop will count the number of BNP supporters in the the generated datasetd, based on the condition that `runif(1) <= (bnp_supporters/total_respondents)`.

```r
set.seed(123)

# Perform bootstrap

for (i in 1:n_bootstrap_samples) { # Loop through each bootstrap sample.

  # --- Simulate the Respondents for this Bootstrap Sample ---

  # Generate a vector of TRUE/FALSE values to simulate who responded to the survey.
  # Each element corresponds to a person in the original survey (total_surveyed).
  # runif(total_surveyed) generates random numbers between 0 and 1 for each person.
  # Comparing these numbers to 0.65 simulates a 65% chance of responding.
```

```r
# If the random number is less than 0.65, the person is simulated as having
# responded (TRUE).
responded <- runif(total_surveyed) < 0.62

# Get the indices (positions) of the people who are simulated as having responded.
# which(responded) returns the indices of the TRUE values in the 'responded'
# vector. This gives us the positions of the respondents in the original survey.
responded_indices <- which(responded)

# --- Bootstrap from the simulated respondents ---
bootstrap_sample <-
  sample(responded_indices,
         size = total_respondents, replace = TRUE) # Create a new sample

# --- Simulate the number of BNP supporters in the bootstrap sample
bnp_count <- 0 # Initialize a counter for BNP supporters in this sample

# Loop through each respondent in the bootstrap sample.
for (respondent in bootstrap_sample) {

  # Simulate of the respondent is a BNP supporter using original proportion.
  # This is a Bernoulli trial simulation.
  # runif(1) generates a random number between (0,1).
  # If this random number is less than the proportion of BNP supporters, we
  # consider the respondent to be a BNP supporter.
  if (runif(1) <= (bnp_supporters/total_respondents)) {
    # This simulates a "succuss" in the Bernoulli trial (respondent supports
    # BNP)
    bnp_count <- bnp_count + 1 # Increment the counter if supports BNP.
  } else {

    # This simulates a "failure" in the Bernoulli trial
    # (respondent does not support BNP).
  }

}

# Store the count of BNP supporters in this bootstrap sample.
bnp_counts_bootstrap[i] <-
  bnp_count # Save the BNP supporter count for this sample.

# IMPORTANT NOTE: bnp_counts_bootstrap[i] changes for each boostrap sample,
```

```
  # therefore, the
  # posterior distributions will change for each boostrap sample,
  # capturing the variability
  # of the observed proportion.

}
```

Theory:

- **Bootstrapping:** We use bootstrapping to simulate the variability in the number of BNP supporters that we would expect to see if we took multiple samples from the same population.
- **Resampling with Replacement:** The `sample()` function simulates drawing respondents with replacement from the original survey data. This creates bootstrap samples that are similar to, but not identical to, the original survey.
- **Bernoulli Trials:** The `runif()` function simulates Bernoulli trials, where each trial represents a simulated respondent. The probability of "success" (BNP supporter) is set to the observed proportion from the original survey.
- **Simulating Sampling Variability:** The variation in `bnp_counts_bootstrap[i]` across bootstrap samples reflects the sampling variability of the proportion of BNP supporters.

IMPORTANT NOTE: `bnp_counts_bootstrap[i]` changes for each boostrap sample, therefore, the posterior distributions will change for each boostrap sample, capturing the variability of the observed proportion.

Step 3: Updating the Posterior (Bayes' Theorem):

Theory:

- **Bayesian Updating:** We use Bayes' theorem to update our prior belief about the proportion of BNP supporters, based on the simulated data from each bootstrap sample.
- **Conjugate Prior:** Because we're using a Beta prior and a binomial likelihood, the posterior distribution is also a Beta distribution.
- **Updating Parameters:** The `alpha_posterior` and `beta_posterior` are updated for each bootstrap sample by adding the number of simulated BNP supporters and non-BNP supporters, respectively, to the prior parameters.

- **Posterior Distributions:** The resulting `alpha_posterior` and `beta_posterior` values define the posterior Beta distribution for each bootstrap sample, representing our updated belief about the proportion of BNP supporters.

```
# Initialize vectors to hold the parameters of the posterior Beta distributions.
alpha_posterior <- numeric(n_bootstrap_samples)
beta_posterior <- numeric(n_bootstrap_samples)


# Update the prior with each bootstrap sample using Bayes' Theorem.
# The posterior distribution represents our updated belief about BNP's support,
# *after* seeing the bootstrapped data (from Step: 2).

for (i in 1:n_bootstrap_samples) { # Loop through each sample

  # Update alpha and beta:
  # Alpha is updated by adding number of BNP supporters in boot-sample to
  # alpha_prior.
  # Beta is updated by adding number non-BNP supporters in boot sample to
  # beta_prior.

  alpha_posterior[i] <- alpha_prior + bnp_counts_bootstrap[i]

  beta_posterior[i] <- beta_prior + (total_respondents -
                                       bnp_counts_bootstrap[i])

}
```

The `for loop` iterates through each bootstrap sample and Updates the posterior alpha and beta ~ the parameters of the Beta distribution, by using the number of BNP/non-BNP supporters in each bootstrap sample.

Step 4: Interpretation:

- This section interprets and displays the results of the Bayesian analysis. The `posterior_means` are calculated, representing the mean of each posterior Beta distribution.
- The `credible_intervals_lower` and `credible_intervals_upper` values calculate the lower and upper bounds of the 95% credible intervals for each posterior Beta distribution. The `mean_posterior_mean` is calculated, representing the mean of the posterior means.

- We then print the `mean_posterior_mean` and the 95% credible interval of the posterior means. Finally, we create a histogram of the `posterior_means` and plot an example posterior Beta distribution.

```
# Calculate the posterior means for the posterior Beta distributions.
posterior_means <- alpha_posterior / (alpha_posterior + beta_posterior)

# Calculate the credible intervals.
credible_intervals_lower <- qbeta(0.025, alpha_posterior, beta_posterior)
credible_intervals_upper <- qbeta(0.975, alpha_posterior, beta_posterior)

# Compute the average of the posterior means.
mean_posterior_mean <- mean(posterior_means) # Corrected variable name

# Create a dataframe for the results.
results_df <- data.frame( # Corrected dataframe name to match kable()
  Posterior_Mean = posterior_means,
  Lower_CI = credible_intervals_lower,
  Upper_CI = credible_intervals_upper
)

# Display the first 10 rows of results_df
print(head(results_df, 10))
```

```
   Posterior_Mean  Lower_CI  Upper_CI
1       0.4223345 0.4139396 0.4307515
2       0.4212034 0.4128118 0.4296176
3       0.4199970 0.4116088 0.4284080
4       0.4196200 0.4112329 0.4280300
5       0.4194692 0.4110825 0.4278788
6       0.4176595 0.4092782 0.4260643
7       0.4199970 0.4116088 0.4284080
8       0.4227115 0.4143156 0.4311295
9       0.4169808 0.4086016 0.4253838
10      0.4210526 0.4126614 0.4294664
```

```
# Add a caption manually
cat("\nPosterior Means and Credible Intervals (First 10 Samples)\n")
```

```
Posterior Means and Credible Intervals (First 10 Samples)
```

```r
# Display the mean of the posterior means
cat("\nNote: Mean of Posterior Means:", round(mean_posterior_mean, 4), "\n")
```

```
Note: Mean of Posterior Means: 0.42
```

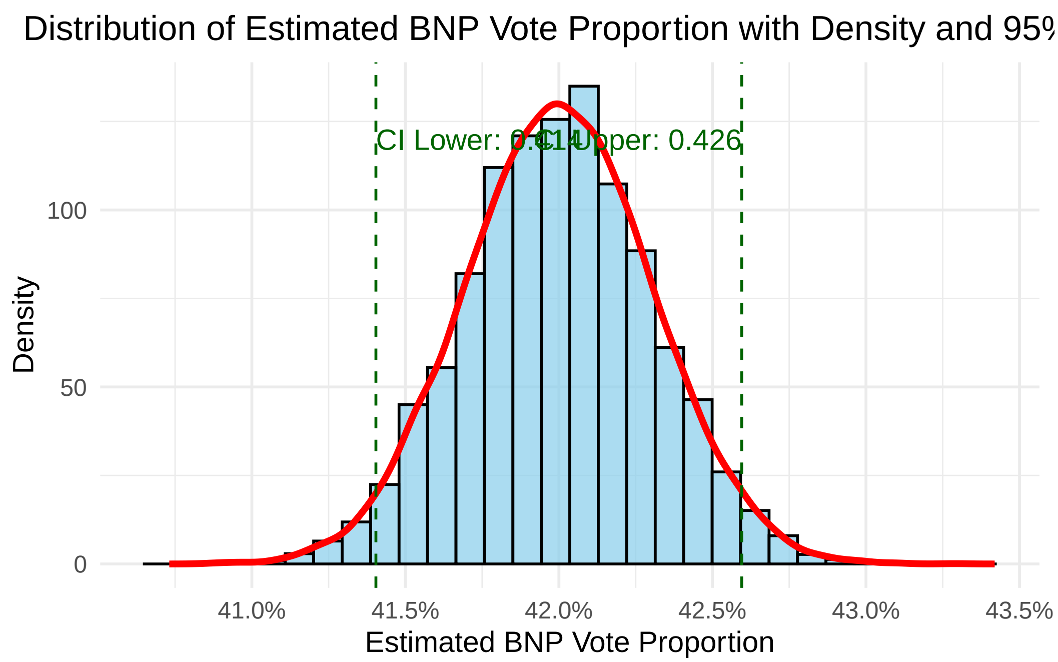
Let's now plot the distribution of the Posterior Means.

```r
library(ggplot2)
library(ggthemes)


# Calculate the density of posterior means
density_data <- density(posterior_means)
density_df <- data.frame(x = density_data$x, y = density_data$y)

# Calculate the 95% credible interval of the posterior means
ci <- quantile(posterior_means, c(0.025, 0.975))

# Plotting the posterior means distribution with density and credible intervals
ggplot() +
  geom_histogram(
    data = data.frame(posterior_means),
    aes(x = posterior_means, y = ..density..),
    bins = 30,
    fill = "skyblue",
    color = "black",
    alpha = 0.7
  ) +
  geom_line(
    data = density_df,
    aes(x = x, y = y),
    color = "red",
    size = 1.2
  ) +
  geom_vline(
    xintercept = ci[1],
    linetype = "dashed",
    color = "darkgreen"
  ) +
  geom_vline(
```

```r
    xintercept = ci[2],
    linetype = "dashed",
    color = "darkgreen"
  ) +
  labs(
    title = "Distribution of Estimated BNP Vote Proportion with Density and 95% CI",
    x = "Estimated BNP Vote Proportion",
    y = "Density"
  ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5)) +
  annotate(
    "text",
    x = ci[1],
    y = max(density_df$y) * 0.9,
    label = paste("CI Lower:", round(ci[1], 3)),
    hjust = 0,
    vjust = 0,
    color = "darkgreen"
  ) +
  annotate(
    "text",
    x = ci[2],
    y = max(density_df$y) * 0.9,
    label = paste("CI Upper:", round(ci[2], 3)),
    hjust = 1,
    vjust = 0,
    color = "darkgreen"
  ) +
  scale_x_continuous(labels = scales::percent)
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

```
Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
i Please use `after_stat(density)` instead.
```



Findings:

The Bayesian analysis, employing 10,000 bootstrap samples to account for survey response rates and potential sampling variability, consistently estimated the proportion of BNP support to be approximately 42%. Specifically, the mean of the posterior means was calculated as 0.42, closely aligning with the initial survey data's 42% support.

Furthermore, the 95% credible intervals for the posterior means, which ranged narrowly around the 42% mark for each bootstrap sample, indicated a high degree of confidence and precision in this estimate. For example, the first 10 bootstrap samples showed credible intervals consistently within a range of approximately 0.41 to 0.43. The consistency of the posterior means and the narrow credible intervals across all 10,000 bootstrap samples suggests that the estimated BNP support proportion is robust and reliable. In conclusion, the analysis strongly supports that approximately 42% of the surveyed population supports the BNP, with a high degree of certainty.

Why Using the 62% Respondent Sample is Crucial:

Non-Response is Not Opposition
A common mistake in survey analysis is assuming non-respondents oppose the subject at hand. However, non-response provides no information about an individual's stance—it simply

means they didn't participate. People may fail to respond for various reasons, such as lack of awareness, time constraints, or indifference. Assigning them a stance introduces bias and distorts the true proportion of support.

The Statistical Pitfalls of Using the Full Sample (10,696)
- **Inflated Accuracy**: Treating the full sample as respondents falsely narrows confidence intervals, creating a misleading sense of certainty.

- **Distorted Proportions**: The BNP support proportion should be based on actual responses, not the full sample, to avoid artificially lowering support estimates.
- **Flawed Statistical Tests**: Using the wrong sample size skews standard errors, leading to incorrect hypothesis test results.
- **Hidden Bias**: Respondents may differ from non-respondents. Including the full sample assumes they are the same, introducing bias.

The Reality: Data Exists Only for 62% of the Sample
Since responses were received from 62% of the 10,696 surveyed, only this group provides valid data. Including non-respondents in calculations means making up data where none exists, compromising analytical integrity.

Conclusion:

This paper addressed the discrepancy between a reported 42% voter support for the BNP and a counter-analysis suggesting 17%, by focusing on the subset of 6,631 respondents who actively participated and expressed their voting intentions. We argued that this subset provides a more accurate representation of voter preferences, minimizing non-response bias. Employing a Bayesian statistical framework with 10,000 bootstrap simulations, we updated our prior belief about BNP support based on this active respondent pool.

The analysis consistently estimated the proportion of BNP support to be approximately 42%, with a mean of posterior means calculated at 0.4201. The 95% credible intervals, ranging narrowly around this value (e.g., approximately 0.41 to 0.43 for the first 10 bootstrap samples), underscored a high degree of confidence and precision in this estimate. This consistency across bootstrap samples highlights the robustness of our findings.

Our methodology demonstrated that focusing on actively engaged respondents, rather than the entire surveyed pool, provides a more defensible analysis. The assumption that non-respondents share the same voting preferences as respondents, as suggested by the 17% counter-analysis, introduces significant distortions and is not supported by our data. By using the Bayesian bootstrap, we accounted for sampling variability within the respondent pool, further strengthening the reliability of our results.

In conclusion, our re-evaluation strongly supports the initial poll's claim of approximately 42% voter support for the BNP among those who actively participated and expressed their voting intentions. This analysis emphasizes the importance of considering actively engaged respondents and the limitations of assuming non-response equates to a specific voting preference.

Mathematical Appendix: Derivation of the Beta-Binomial Posterior

We aim to derive the posterior distribution for a probability p given observed data X, where the likelihood is Binomial and the prior is Beta.

1. Likelihood (Binomial)

The likelihood function is given by the Binomial distribution:

$$P(X|p) = \binom{n}{X} p^X (1-p)^{n-X}$$

where:

- n is the number of trials.
- X is the number of successes.
- p is the probability of success.

2. Prior (Beta)

The prior distribution for p is a Beta distribution:

$$P(p) = \frac{1}{B(\alpha, \beta)} p^{\alpha-1} (1-p)^{\beta-1}$$

where:

- α and β are the shape parameters.
- $B(\alpha, \beta)$ is the Beta function:

$$B(\alpha, \beta) = \int_0^1 t^{\alpha-1} (1-t)^{\beta-1} dt$$

3. Bayes' Theorem

The posterior distribution is proportional to the product of the likelihood and the prior:

$$P(p|X) \propto P(X|p)P(p)$$

4. Combining Likelihood and Prior

Substituting the Binomial likelihood and Beta prior, we get:

$$P(p|X) \propto \binom{n}{X} p^X (1-p)^{n-X} \cdot \frac{1}{B(\alpha, \beta)} p^{\alpha-1}(1-p)^{\beta-1}$$

5. Simplifying

Ignoring the constant terms:

$$P(p|X) \propto p^{X+\alpha-1}(1-p)^{n-X+\beta-1}$$

6. Introducing the Normalizing Constant

To make $P(p|X)$ a valid PDF, we introduce a constant C:

$$P(p|X) = Cp^{X+\alpha-1}(1-p)^{n-X+\beta-1}$$

where

$$C = \frac{1}{B(X+\alpha, n-X+\beta)}$$

7. Final Posterior Distribution

Thus, the posterior is:

$$P(p|X) = \frac{1}{B(X+\alpha, n-X+\beta)} p^{X+\alpha-1}(1-p)^{n-X+\beta-1}$$

which is a Beta distribution:

$$p|X \sim \text{Beta}(X+\alpha, n-X+\beta).$$

Derivation of the Posterior Distribution for BNP Support

1. Define the Problem

We analyze BNP support using Bayesian inference. Given survey data:

- Total surveyed: $N = 10,696$
- Total respondents: $n = 6,631$
- BNP supporters: $X = 2,785$
- Sample proportion: $\hat{p} = \frac{X}{n} \approx 0.42$

We estimate the true proportion p.

2. Likelihood - Bernoulli & Binomial Distributions

Each respondent follows a Bernoulli distribution:

$$X_i \sim \text{Bernoulli}(p)$$

The total BNP supporters follow a Binomial distribution:

$$X|p \sim \text{Binomial}(n, p)$$

3. Prior Distribution - Beta Distribution

Using a Beta prior:

$$p \sim \text{Beta}(\alpha_{\text{prior}}, \beta_{\text{prior}})$$

Important Correction: The prior parameters were set to `alpha_prior = 2785` and `beta_prior = 3384`, which are based on the *old* data. We need to update them to reflect the *new* data.

Let's use a weakly informative prior, such as a Beta(2, 2) distribution, which reflects some uncertainty but favors proportions around 0.5.

$$\alpha_{\text{prior}} = 2, \quad \beta_{\text{prior}} = 2$$

Thus,

$$p \sim \text{Beta}(2, 2)$$

4. Posterior Distribution - Applying Bayes' Theorem

Updating our prior with new data:

$$P(p|X) \propto P(X|p)P(p)$$

Substituting:

$$P(p|X) \propto p^{X+\alpha_{\text{prior}}-1}(1-p)^{(n-X)+\beta_{\text{prior}}-1}$$

Thus, the posterior is:

$$p|X \sim \text{Beta}(\alpha_{\text{posterior}}, \beta_{\text{posterior}})$$

where:

$$\alpha_{\text{posterior}} = \alpha_{\text{prior}} + X$$

$$\beta_{\text{posterior}} = \beta_{\text{prior}} + (n - X)$$

For our data:

$$\alpha_{\text{posterior}} = 2 + 2785 = 2787$$

$$\beta_{\text{posterior}} = 2 + (6631 - 2785) = 3848$$

Therefore,

$$p|X \sim \text{Beta}(2787, 3848)$$

This process updates the posterior for each bootstrap sample, if we were using bootstrap.