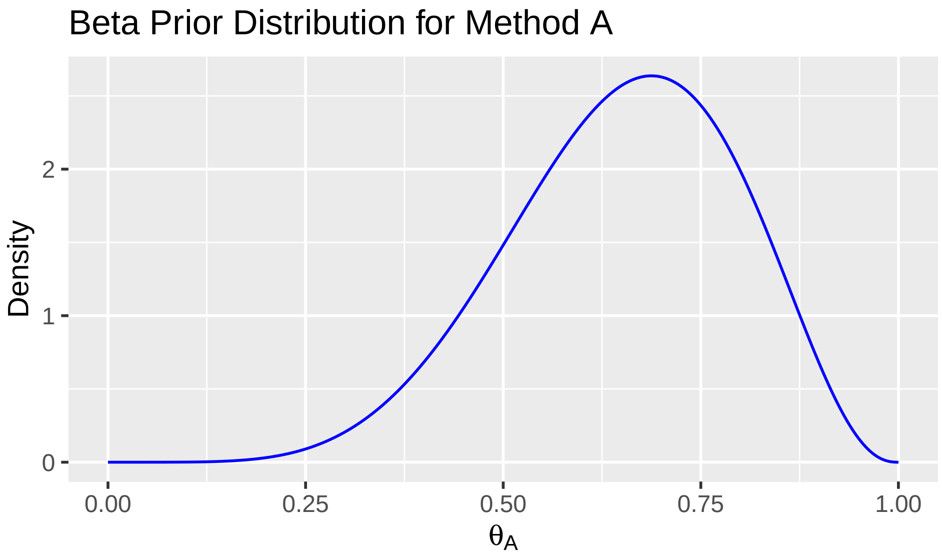
STAT40850 -Bayesian Analysis Assignment 1

Isha Borgaonkar (24209758)

Question 1: 1.A) Specify and plot a Beta prior distribution for **method A**.

```
library(ggplot2)
suppressMessages(suppressWarnings(library(extraDistr)))
mean_theta_A <- 0.65 # Given mean
total_counts <- 10  # Adjust as necessary
alpha_prior <- mean_theta_A * total_counts
beta_prior <- (1 - mean_theta_A) * total_counts
theta_vals <- seq(0, 1, length.out = 1000) # Plot Beta Prior
beta_prior_vals <- dbeta(theta_vals, alpha_prior, beta_prior)
ggplot(data.frame(theta = theta_vals, density = beta_prior_vals), aes(x = theta, y = density)
  geom_line(color = "blue") +
  ggtitle("Beta Prior Distribution for Method A") +
  xlab(expression(theta[A])) +
  ylab("Density")
```



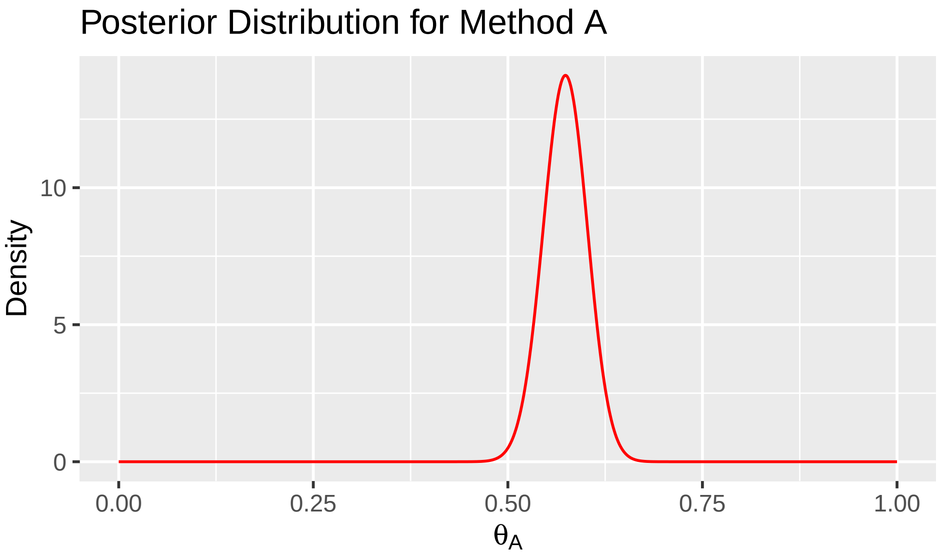
1.B) Calculat prior probability **Pr(theta A $<$ 0.5)**

```r
prob_theta_less_0.5 <- pbeta(0.5, alpha_prior, beta_prior)
cat("The prior probability Pr(theta_A < 0.5) is", round(prob_theta_less_0.5, 4),"\n")
```

```
The prior probability Pr(theta_A < 0.5) is 0.159
```

Question 2: 2.A)Estimate and plot the posterior distribution for method A.

```r
n_A <- 296 # Given data
x_A <- 169
# Posterior parameters
alpha_post <- alpha_prior + x_A
beta_post <- beta_prior + (n_A - x_A)
# Plot Beta Posterior
theta_post_vals <- dbeta(theta_vals, alpha_post, beta_post)
ggplot(data.frame(theta = theta_vals, density = theta_post_vals), aes(x = theta, y = density)
  geom_line(color = "red") +
  ggtitle("Posterior Distribution for Method A") +
  xlab(expression(theta[A])) +
  ylab("Density")
```



2.B) Compute posterior probability Pr(theta A $>$ 0.7)

```r
prob_theta_greater_0.7 <- 1 - pbeta(0.7, alpha_post, beta_post)
cat("The posterior probability Pr(theta_A > 0.7) is", round(prob_theta_greater_0.7, 4),"\n")
```

```
The posterior probability Pr(theta_A > 0.7) is 0
```

Question 3: Present a brief commentary on the results obtained focusing on whether there is any evidence against the hypothesis that theta = 0.6. a posteriori.

```r
# Compute posterior mean
mean_theta_post <- alpha_post / (alpha_post + beta_post)
cat("Posterior mean for theta_A:", round(mean_theta_post, 4), "\n")
```

```
Posterior mean for theta_A: 0.5735
```

```r
# Compute posterior probability P( _A > 0.60)
prob_theta_greater_0.60 <- 1 - pbeta(0.60, alpha_post, beta_post)
cat("Probability P( _A > 0.60):", round(prob_theta_greater_0.60, 4), "\n")
```

```
Probability P( _A > 0.60): 0.1748
```

```r
prob_theta_less_0.60 <- pbeta(0.60, alpha_post, beta_post)
cat("Probability P( _A < 0.60):", round(prob_theta_less_0.60, 4),"\n")
```

```
Probability P( _A < 0.60): 0.8252
```

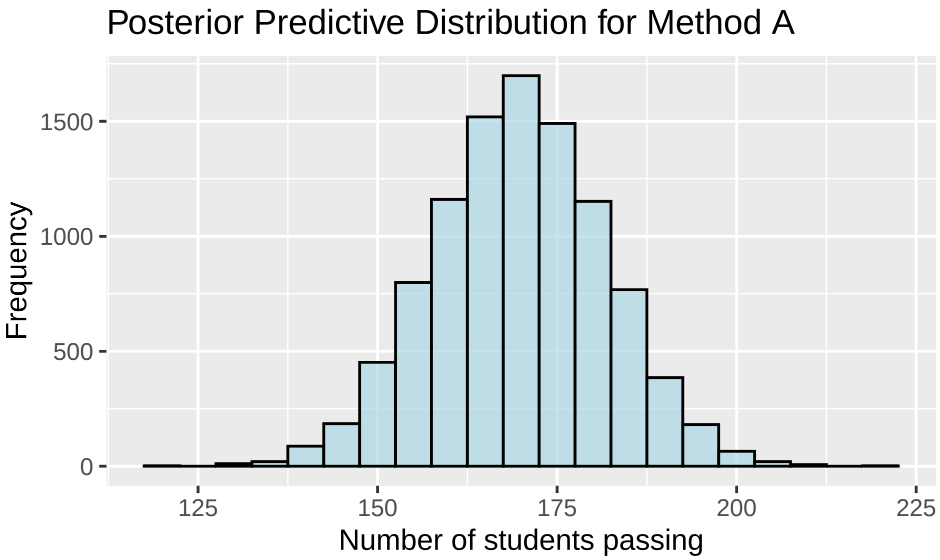
- P(theta A < 0.60) = 0.8252 • 82.52% of the posterior distribution suggests that theta A is less than 0.60.

- P(theta A > 0.60) = 0.1748 • 17.48% of the posterior distribution suggests that theta A is greater than 0.60

Since P(theta A > 0.60) is relatively small, there is little evidence to support the hypothesis that theta A = 0.60 based on the data. The posterior distribution shows that values less than 0.60 are much more probable, and therefore, we could reasonably conclude that the true value of theta A is unlikely to be exactly 0.60 a posteriori.Thus, the data suggests that theta A is most likely below 0.60, which provides evidence against the hypothesis that theta A = 0.6.

Question 4: Estimate (via Monte Carlo sampling) and plot the posterior predictive distribution Comment on the fit of the model to the observed data. Use the sample simulated from the posterior predictive distribution to estimate the probability.

```r
# Monte Carlo Sampling for Predictive Distribution
set.seed(123)
num_samples <- 10000
posterior_samples <- rbeta(num_samples, alpha_post, beta_post)
predictive_samples <- rbinom(num_samples, n_A, posterior_samples)
ggplot(data.frame(x_tilde = predictive_samples), aes(x = x_tilde)) +
  geom_histogram(binwidth = 5, fill = "lightblue", alpha = 0.7, color = "black") +
  ggtitle("Posterior Predictive Distribution for Method A") +
  xlab("Number of students passing") +
  ylab("Frequency")
```



```r
## Compute probability Pr(~xA   180 | xA)
prob_x_tilde_geq_180 <- mean(predictive_samples >= 180)
prob_x_tilde_geq_180
```

```
[1] 0.2101
```

The histogram is of predictive distribution for the number of students passing the exam. The probability Pr(~xA   180 | xA) is 0.2101 • The estimated probability Pr(~xA   180 | xA) =0.21019 means that about 21.01% of the simulated posterior predictive values are greater than or equal to 180. • Good but not perfect fit: The model is reasonable and captures the overall trend, but it may slightly underestimate the probability of higher counts.

Question 5: Use Stan to estimate the posterior distribution for the probability of passing the exam for each group (theta A and theta B) using the same prior adopted above for theta A and a symmetric prior for theta B. Estimate the posterior distribution of the difference theta diff = theta B − theta A.

```r
library(rstan)
```

```
Warning: package 'rstan' was built under R version 4.4.2
```

```
Loading required package: StanHeaders
```

```
Warning: package 'StanHeaders' was built under R version 4.4.2
```

```

rstan version 2.32.6 (Stan version 2.32.2)
```

```
For execution on a local, multicore CPU with excess RAM we recommend calling
options(mc.cores = parallel::detectCores()).
To avoid recompilation of unchanged Stan programs, we recommend calling
rstan_options(auto_write = TRUE)
For within-chain threading using `reduce_sum()` or `map_rect()` Stan functions,
change `threads_per_chain` option:
rstan_options(threads_per_chain = 1)
```

```
Do not specify '-march=native' in 'LOCAL_CPPFLAGS' or a Makevars file
```

```r
# Define data for Stan
stan_data <- list(
  n_A = 296, x_A = 169,
  n_B = 380, x_B = 247,
  alpha_prior = 13, beta_prior = 7
)
# Stan model for Bayesian estimation
stan_model_code <- "
  data {
    int<lower=0> n_A;
    int<lower=0> x_A;
    int<lower=0> n_B;
    int<lower=0> x_B;
    real<lower=0> alpha_prior;
    real<lower=0> beta_prior;
  }
  parameters {
    real<lower=0,upper=1> theta_A;
    real<lower=0,upper=1> theta_B;
  }
  model {
    theta_A ~ beta(alpha_prior, beta_prior);
    theta_B ~ beta(alpha_prior, beta_prior);
    x_A ~ binomial(n_A, theta_A);
    x_B ~ binomial(n_B, theta_B);
  }
  generated quantities {
    real theta_diff = theta_B - theta_A;
  }
"
# Compile and run Stan model
fit <- stan(model_code = stan_model_code, data = stan_data, iter=4000,
            warmup=1000, chains=4, thin=1, refresh=0, verbose=FALSE)
print(fit, pars=c("theta_A", "theta_B", "theta_diff"), probs=c(0.025,
                            0.25, 0.5, 0.75, 0.975), digits_summary=2)
```

```
Inference for Stan model: anon_model.
4 chains, each with iter=4000; warmup=1000; thin=1;
post-warmup draws per chain=3000, total post-warmup draws=12000.

           mean se_mean   sd 2.5%  25%  50%  75% 97.5% n_eff Rhat
theta_A    0.58       0 0.03 0.52 0.56 0.58 0.59  0.63  9113    1
theta_B    0.65       0 0.02 0.60 0.63 0.65 0.67  0.70 11071    1
theta_diff 0.07       0 0.04 0.01 0.05 0.07 0.10  0.15  9588    1

Samples were drawn using NUTS(diag_e) at Wed Feb 12 20:40:30 2025.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

```
cat("Model has completed successfully.\n")
```

```
Model has completed successfully.
```

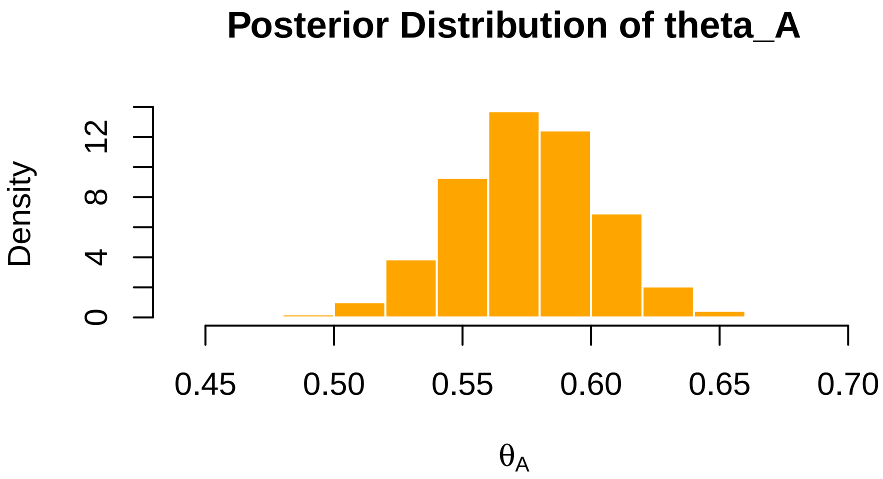
This code uses Stan to estimate the posterior distributions for the probability of passing the exam for both Method A (theta A) and Method B theta B).It assumes a Beta prior for both parameters and models the number of students passing as binomially distributed. The code also computes theta diff=theta B − theta A. to compare the effectiveness of both methods. Finally, it compiles and runs the Bayesian model using Stan with 2000 iterations and 4 chains.

Question 6: Plot and summarise the posterior distributions estimated in the previous question; estimate of the posterior probability Pr(theta diff < A, |x B) by using the Stan MCMC output; present a brief commentary on the results obtained.

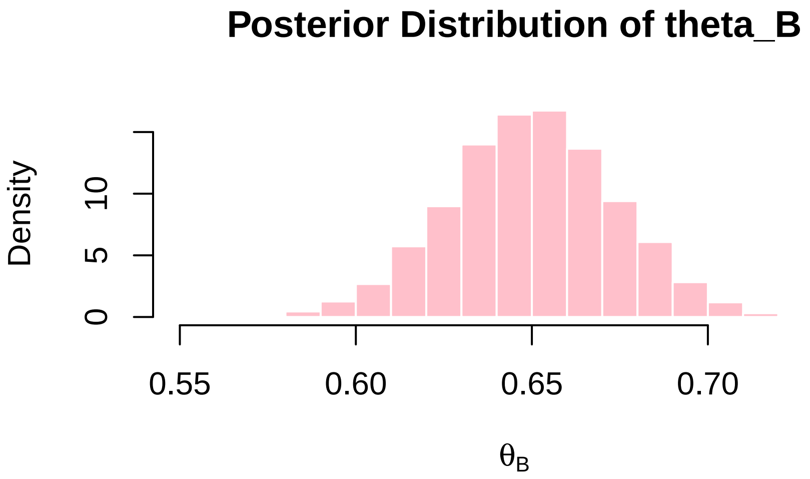
```r
# Ensure fit object is valid
if (!exists("fit")) {
  stop("Error: The 'fit' object does not exist. Run the Stan model first.")
}
# Extract posterior samples
posterior_samples <- as.array(fit)
# Extract individual parameter samples
theta_A_samples <- posterior_samples[, , "theta_A"]
theta_B_samples <- posterior_samples[, , "theta_B"]
theta_diff_samples <- posterior_samples[, , "theta_diff"]
# Compute probability P( diff < 0 | xA, xB)
p_theta_diff_lt_0 <- mean(theta_diff_samples < 0)
print(p_theta_diff_lt_0)
```

```
[1] 0.01691667
```

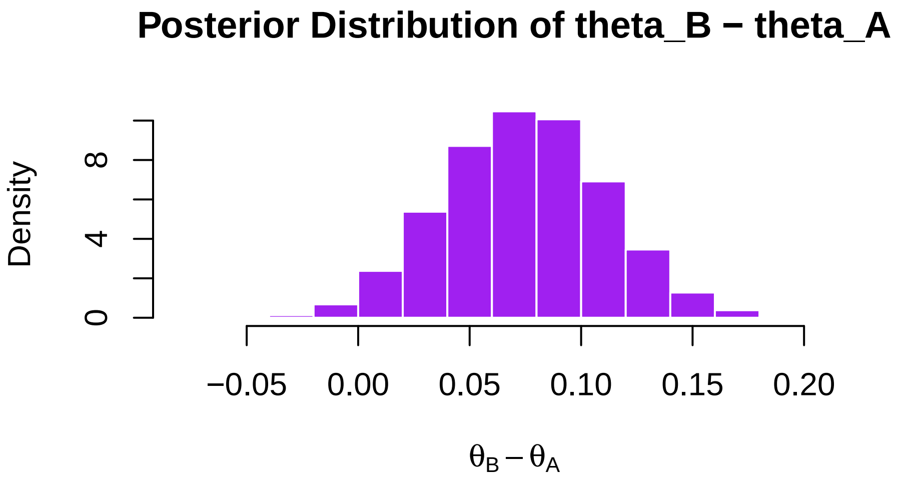
```r
# Plot posterior distributions
hist(theta_A_samples, col="orange", border="white",
     main="Posterior Distribution of theta_A",
     xlab=expression(theta[A]), probability=TRUE)
```



```r
hist(theta_B_samples, col="pink", border="white",
     main="Posterior Distribution of theta_B",
     xlab=expression(theta[B]), probability=TRUE)
```



```r
hist(theta_diff_samples, col="purple", border="white",
     main="Posterior Distribution of theta_B - theta_A",
     xlab=expression(theta[B] - theta[A]), probability=TRUE)
```



Interpretation: 1)The posterior distributions of **A** and **B** allow us to compare the effectiveness of both methods. 2)If **P(diff < 0)** is high, it suggests that Method A is likely better than Method B. 3)The histogram visualizations help in understanding the spread and uncertainty of each parameter.