Bayesian A1

Zhengyang Fei

Problem 1

Question 1:

(a)

We will use Baye's Theorem,

$$P(D^+|T^+) = \frac{P(T^+|D^+)P(D^+)}{P(T^+)}$$

where:

- $P(D^+) = 0.13$ prior probability of lupus
- $P(T^+|D^+) = 0.99$ sensitivity of ANA test
- $P(T^-|D^-) = 0.80$ specificity of ANA test

and

$$P(T^+) = P(T^+|D^+)P(D^+) + P(T^+|D^-)P(D^-) = (0.99)(0.13) + (1 - 0.80)(1 - 0.13) = 0.3027$$

```
P_D_positive <- 0.13  # Prior probability of lupus
P_T_positive_given_D_positive <- 0.99  # Sensitivity
P_T_negative_given_D_negative <- 0.8  # Specificity
P_D_negative <- 1 - P_D_positive  # Complement of prevalence
P_T_positive_given_D_negative <- 1 - P_T_negative_given_D_negative  # False positive rate

# P(T+)
P_T_positive <- (P_T_positive_given_D_positive * P_D_positive) +
                (P_T_positive_given_D_negative * P_D_negative)
```

```r
# P(D+|T+)
P_D_positive_given_T_positive <- (P_T_positive_given_D_positive * P_D_positive) / P_T_positiv

cat("Hence the probability is", P_D_positive_given_T_positive, "\n")
```

```
Hence the probability is 0.4251734
```

(b)

We now update the probabilities given the new information,

- $P(D^+|T_1^+) = 0.4252$ (prior probability from (a))
- $P(T_2^+|D^+) = 0.73$ (sensitivity of Anti-dsDNA test)
- $P(T_2^-|D^-) = 0.98$ (specificity of Anti-dsDNA test)

and

$$P(T_2^+) = P(T_2^+|D^+)P(D^+|T_1^+)+P(T_2^+|D^-)P(D^-|T_1^+) = (0.73)(0.4252)+(1-0.98)(1-0.4252) = 0.3219$$

Applying Baye's theorem once more,

$$P(D^+|T_1^+, T_2^+) = \frac{P(T_2^+|D^+)P(D^+|T_1^+)}{P(T_2^+)}$$

```r
# ANA test
P_D_positive <- 0.13  # Prior probability of lupus
P_T1_positive_given_D_positive <- 0.99  # Sensitivity of ANA test
P_T1_negative_given_D_negative <- 0.8  # Specificity of ANA test
P_D_negative <- 1 - P_D_positive  # Complement of prevalence
P_T1_positive_given_D_negative <- 1 - P_T1_negative_given_D_negative  # False positive rate
# P(T1+)
P_T1_positive <- (P_T1_positive_given_D_positive * P_D_positive) +
                 (P_T1_positive_given_D_negative * P_D_negative)
# P(D+|T1+)
P_D_positive_given_T1_positive <- (P_T1_positive_given_D_positive * P_D_positive) / P_T1_posi

# Anti-dsDNA test
P_T2_positive_given_D_positive <- 0.73  # Sensitivity of Anti-dsDNA test
P_T2_negative_given_D_negative <- 0.98  # Specificity of Anti-dsDNA test
```

```r
P_T2_positive_given_D_negative <- 1 - P_T2_negative_given_D_negative  # False positive rate
# Use P(D+ | T1+) as the new prior probability
P_D_positive_new_prior <- P_D_positive_given_T1_positive
P_D_negative_new_prior <- 1 - P_D_positive_new_prior
# Compute P(T2+)
P_T2_positive <- (P_T2_positive_given_D_positive * P_D_positive_new_prior) +
                 (P_T2_positive_given_D_negative * P_D_negative_new_prior)
# P(D+|T1+ and T2+)
P_D_positive_given_T1_T2_positive <- (P_T2_positive_given_D_positive * P_D_positive_new_prio

# Print result
cat("Hence the updated probability after two positive tests is", P_D_positive_given_T1_T2_pos
```

```
Hence the updated probability after two positive tests is 0.9642824
```

(c)

In the first stage, a good screening test should have high sensitivity $P(T^+|D^+)$, meaning it correctly identifies most patients who actually have the disease. This minimizes false negatives, ensuring that few true cases go undiagnosed. However, it is also important that the specificity is not too low because a very low specificity would result in too many false positives, leading to an excessive number of unnecessary confirmatory tests.

In the second stage, a good confirmatory test should have high specificity $P(T^-|D^-)$, meaning it accurately rules out individuals who do not have the disease. This minimizes false positives, ensuring that patients who tested positive in the screening test truly have the disease and are not misdiagnosed.
By combining a high-sensitivity screening test with a high-specificity confirmatory test, we achieve an effective diagnostic process, minimizing both false negatives (missed cases) and false positives (incorrect diagnoses).

(d)

```r
P_D_positive_male <- 1 / 25000  # Prevalence of lupus in males
P_T_positive_given_D_positive <- 0.99  # Sensitivity of ANA test
P_T_negative_given_D_negative <- 0.80  # Specificity of ANA test
P_D_negative_male <- 1 - P_D_positive_male  # Complement of prevalence
P_T_positive_given_D_negative <- 1 - P_T_negative_given_D_negative  # False positive rate

# Compute P(T+)
```

```r
P_T_positive_male <- (P_T_positive_given_D_positive * P_D_positive_male) +
                     (P_T_positive_given_D_negative * P_D_negative_male)

# Compute P(D+ | T+)
P_D_positive_given_T_positive_male <- (P_T_positive_given_D_positive * P_D_positive_male) / P

# Print result
cat("Hence the probability that a male who tested positive actually has lupus is", P_D_positi
```

```
Hence the probability that a male who tested positive actually has lupus is 0.0001979687
```

Problem 2

(a)

Prior 1

- Likelihood model: $X_i \sim N(\mu, \tau^{-1})$ with $\tau = \frac{1}{36}$ (variance = 36).
- Prior distribution: $\mu \sim N(\mu_0, \tau_0^{-1})$ with:
  - $\mu_0 = 66$ (prior mean).
  - $\tau_0 = \frac{4}{9}$ (prior precision, so prior variance = $\frac{9}{4}$).

```
mu_0 <- 66
tau_0 <- 4/9  # fixed rior precision
sigma_0 <- sqrt(1/tau_0)  # Prior standard deviation

# Given likelihood parameters
tau <- 1/36  # Precision of the likelihood
sigma <- sqrt(1/tau)  # Standard deviation of individual observations

D1 <- c(53, 49, 63, 72, 55, 65)
D2 <- c(28, 27, 36, 42, 25, 35)

compute_posterior_t <- function(data, mu_0, tau_0, tau) {
  n <- length(data)
  x_bar <- mean(data)

  # posterior mean and precision
  tau_post <- tau_0 + n * tau
  mu_post <- (tau_0 * mu_0 + n * tau * x_bar) / tau_post
  sigma_post <- sqrt(1 / tau_post)  # Posterior se

  #t_critical <- qt(0.975, df=n-1)
  ci_lower <- mu_post - 1.96 * sigma_post
  ci_upper <- mu_post + 1.96 * sigma_post

  return(list(
    "Posterior Mean" = mu_post,
    "Posterior Std Dev" = sigma_post,
    "95% Credible Interval" = c(ci_lower, ci_upper)
  ))
}
```

```r
posterior_t_D1 <- compute_posterior_t(D1, mu_0, tau_0, tau)
posterior_t_D2 <- compute_posterior_t(D2, mu_0, tau_0, tau)

# Print results
print("Posterior results for D1 (Analytical using t-distribution):")
```

```
[1] "Posterior results for D1 (Analytical using t-distribution):"
```

```r
print(posterior_t_D1)
```

```
$`Posterior Mean`
[1] 64.22727

$`Posterior Std Dev`
[1] 1.279204

$`95% Credible Interval`
[1] 61.72003 66.73451
```

```r
print("Posterior results for D2 (Analytical using t-distribution):")
```

```
[1] "Posterior results for D2 (Analytical using t-distribution):"
```

```r
print(posterior_t_D2)
```

```
$`Posterior Mean`
[1] 56.77273

$`Posterior Std Dev`
[1] 1.279204

$`95% Credible Interval`
[1] 54.26549 59.27997
```

Prior 2

```r
library(MCMCpack)  # for inverse-gamma sampling
```

```
Loading required package: coda
```

```
Loading required package: MASS
```

```
##
## Markov Chain Monte Carlo Package (MCMCpack)
```

```
## Copyright (C) 2003-2025 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
```

```
##
## Support provided by the U.S. National Science Foundation
```

```
## (Grants SES-0350646 and SES-0350613)
##
```

```r
mu_0 <- 66
theta <- 4
alpha <- 1
beta <- 25

D1 <- c(53, 49, 63, 72, 55, 65)
D2 <- c(28, 27, 36, 42, 25, 35)

bayesian_simulation_prior2 <- function(data, mu_0, alpha, beta, theta, num_samples=10000) {
  n <- length(data)
  x_bar <- mean(data)

  # Compute posterior parameters
  alpha_post <- alpha + n / 2
  beta_post <- beta + 0.5 * sum((data - x_bar)^2) + (theta * n * (x_bar - mu_0)^2) / (2 * (t

  # Monte Carlo sampling for tau (precision)
  tau_samples <- rgamma(num_samples, shape = alpha_post, rate = beta_post)

  # Monte Carlo sampling for mu given sampled tau
  mu_samples <- rnorm(num_samples,
                      mean = (theta * mu_0 + n * x_bar) / (theta + n),
```

```
                      sd = sqrt(1 / ((theta + n) * tau_samples)))

  post_mean <- mean(mu_samples)
  post_sd <- sd(mu_samples)

  ci_lower <- quantile(mu_samples, 0.025)
  ci_upper <- quantile(mu_samples, 0.975)

  return(list(
    "Posterior Mean" = post_mean,
    "Posterior Std Dev" = post_sd,
    "95% Credible Interval" = c(ci_lower, ci_upper)
  ))
}

posterior_sim_D1 <- bayesian_simulation_prior2(D1, mu_0, alpha, beta, theta)
posterior_sim_D2 <- bayesian_simulation_prior2(D2, mu_0, alpha, beta, theta)

print("Posterior results for D1 (Prior 2):")
```

```
[1] "Posterior results for D1 (Prior 2):"
```

```
print(posterior_sim_D1)
```

```
$`Posterior Mean`
[1] 62.11074

$`Posterior Std Dev`
[1] 2.928472

$`95% Credible Interval`
    2.5%    97.5%
56.15181 67.88347
```

```
print("Posterior results for D2 (Prior 2):")
```

```
[1] "Posterior results for D2 (Prior 2):"
```

```r
print(posterior_sim_D2)
```

```
$`Posterior Mean`
[1] 45.75438

$`Posterior Std Dev`
[1] 7.122892

$`95% Credible Interval`
    2.5%    97.5%
31.29539 59.67038
```

Prior 3

```r
mu_0 <- 66
alpha <- 0.001
beta <- 0.001
theta <- 0.1

D1 <- c(53, 49, 63, 72, 55, 65)
D2 <- c(28, 27, 36, 42, 25, 35)

bayesian_simulation_prior3 <- function(data, mu_0, alpha, beta, theta, num_samples=10000) {
  n <- length(data)
  x_bar <- mean(data)

  # Compute posterior parameters
  alpha_post <- alpha + n / 2
  beta_post <- beta + 0.5 * sum((data - x_bar)^2) + (theta * n * (x_bar - mu_0)^2) / (2 * (t

  # Monte Carlo sampling for tau (precision)
  tau_samples <- rgamma(num_samples, shape = alpha_post, rate = beta_post)

  # Monte Carlo sampling for mu given sampled tau
  mu_samples <- rnorm(num_samples,
                      mean = (theta * mu_0 + n * x_bar) / (theta + n),
                      sd = sqrt(1 / ((theta + n) * tau_samples)))

  post_mean <- mean(mu_samples)
  post_sd <- sd(mu_samples)
```

```r
  ci_lower <- quantile(mu_samples, 0.025)
  ci_upper <- quantile(mu_samples, 0.975)

  return(list(
    "Posterior Mean" = post_mean,
    "Posterior Std Dev" = post_sd,
    "95% Credible Interval" = c(ci_lower, ci_upper)
  ))
}

posterior3_sim_D1 <- bayesian_simulation_prior3(D1, mu_0, alpha, beta, theta)
posterior3_sim_D2 <- bayesian_simulation_prior3(D2, mu_0, alpha, beta, theta)

print("Posterior results for D1 (Prior 3):")
```

```
[1] "Posterior results for D1 (Prior 3):"
```

```r
print(posterior3_sim_D1)
```

```
$`Posterior Mean`
[1] 59.59692

$`Posterior Std Dev`
[1] 3.918122

$`95% Credible Interval`
    2.5%    97.5%
51.61361 67.33273
```

```r
print("Posterior results for D2 (Prior 3):")
```

```
[1] "Posterior results for D2 (Prior 3):"
```

```r
print(posterior3_sim_D2)
```

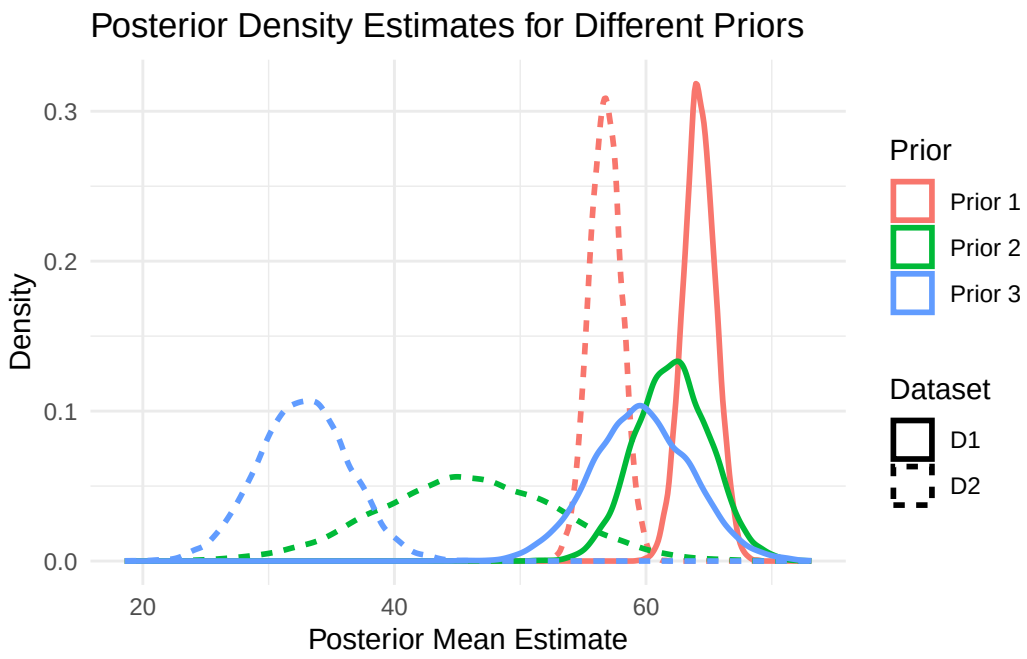
```
$`Posterior Mean`
[1] 32.75506
```

```
$`Posterior Std Dev`
[1] 3.642207

$`95% Credible Interval`
    2.5%    97.5%
25.60985 40.03058
```

(b)

```r
library(ggplot2)

# Prior 1
prior1_D1 <- rnorm(10000, mean=posterior_t_D1$`Posterior Mean`, sd=posterior_t_D1$`Posterior
prior1_D2 <- rnorm(10000, mean=posterior_t_D2$`Posterior Mean`, sd=posterior_t_D2$`Posterior

# Prior 2, 3
prior2_D1 <- rnorm(10000, mean=posterior_sim_D1$`Posterior Mean`, sd=posterior_sim_D1$`Poste
prior2_D2 <- rnorm(10000, mean=posterior_sim_D2$`Posterior Mean`, sd=posterior_sim_D2$`Poste
prior3_D1 <- rnorm(10000, mean=posterior3_sim_D1$`Posterior Mean`, sd=posterior3_sim_D1$`Pos
prior3_D2 <- rnorm(10000, mean=posterior3_sim_D2$`Posterior Mean`, sd=posterior3_sim_D2$`Pos

posterior_data <- data.frame(
  value = c(prior1_D1, prior1_D2, prior2_D1, prior2_D2, prior3_D1, prior3_D2),
  dataset = rep(c("D1", "D2", "D1", "D2", "D1", "D2"), each = 10000),
  prior = rep(c("Prior 1", "Prior 1", "Prior 2", "Prior 2", "Prior 3", "Prior 3"), each = 100
)

ggplot(posterior_data, aes(x = value, color = prior, linetype = dataset)) +
  geom_density(size = 1) +
  labs(title = "Posterior Density Estimates for Different Priors",
       x = "Posterior Mean Estimate",
       y = "Density",
       color = "Prior",
       linetype = "Dataset") +
  theme_minimal()
```

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

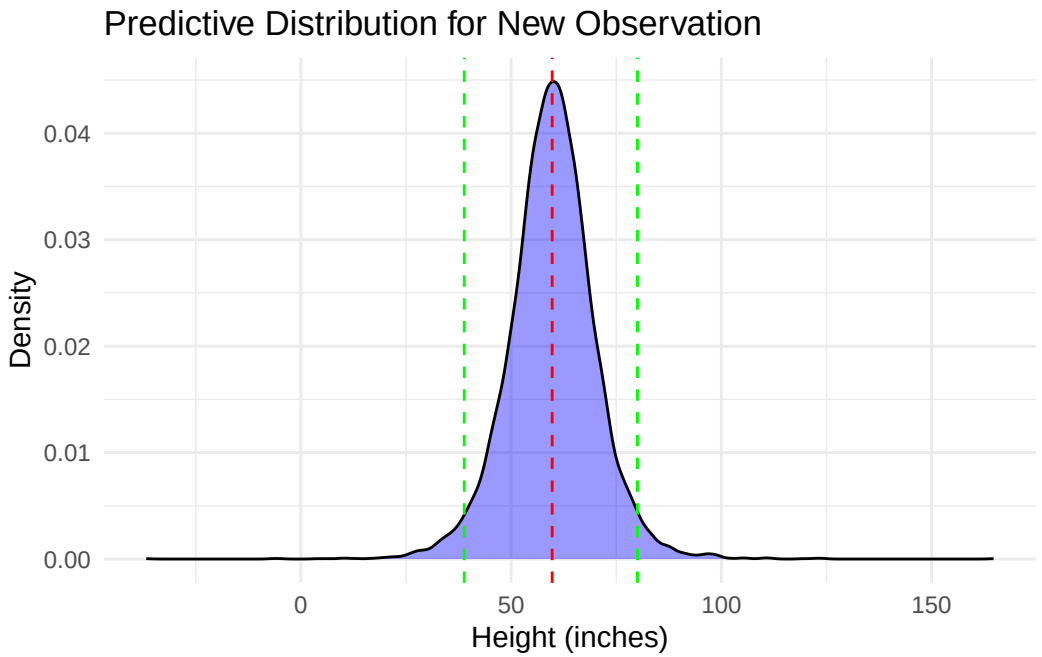


(c)

```
set.seed(2025)
D1 <- c(53, 49, 63, 72, 55, 65)
mu_0 <- 66
theta <- 0.1
alpha <- 0.001
beta <- 0.001

bayesian_predictive_prior3 <- function(data, mu_0, alpha, beta, theta, num_samples=10000) {
  n <- length(data)
  x_bar <- mean(data)

  alpha_post <- alpha + n / 2
  beta_post <- beta + 0.5 * sum((data - x_bar)^2) + (theta * n * (x_bar - mu_0)^2) / (2 * (t

  tau_samples <- rgamma(num_samples, shape = alpha_post, rate = beta_post)

  mu_samples <- rnorm(num_samples,
                      mean = (theta * mu_0 + n * x_bar) / (theta + n),
                      sd = sqrt(1 / ((theta + n) * tau_samples)))
```

```r
  pred_samples <- rnorm(num_samples, mean = mu_samples, sd = sqrt(1 / tau_samples))

  pred_mean <- mean(pred_samples)
  pred_sd <- sd(pred_samples)
  cred_int <- quantile(pred_samples, c(0.025, 0.975))

  cat("Predictive Mean:", pred_mean, "\n")
  cat("Predictive Std Dev:", pred_sd, "\n")
  cat("95% Credible Interval:", cred_int[1], "-", cred_int[2], "\n")

  plot <- ggplot(data = data.frame(x = pred_samples), aes(x = x)) +
    geom_density(fill = "blue", alpha = 0.4) +
    geom_vline(xintercept = pred_mean, color = "red", linetype = "dashed") +
    geom_vline(xintercept = cred_int, color = "green", linetype = "dashed") +
    labs(title = "Predictive Distribution for New Observation",
         x = "Height (inches)",
         y = "Density") +
    theme_minimal()

  print(plot)

  return(list(
    "Predictive Mean" = pred_mean,
    "Predictive Std Dev" = pred_sd,
    "95% Credible Interval" = c(cred_int[1], cred_int[2]),
    "mu_samples" = mu_samples,
    "tau_samples" = tau_samples
  ))
}

results <- bayesian_predictive_prior3(D1, mu_0, alpha, beta, theta)
```

```
Predictive Mean: 59.76416
Predictive Std Dev: 10.38613
95% Credible Interval: 38.8518 - 80.04
```



(d)

Some general observation on how priors differ

- Prior 1: The model assumes

$$X_i \sim N(\mu, \tau^{-1})$$

$$\mu \sim N(\mu_0 = 66, \tau_0^{-1} = 4/9)$$

  Since τ_0 is fixed, the posterior mean is strongly influenced by the prior, making it less adaptive to the data. Hence the posterior mean stays close to $\mu_0 = 66$ inches and the standard deviation is almost fixed, hence leading to a tight distribution. Flexibility is also low due to this reason.

- Prior 2: The model assumes

$$X_i|\mu, \tau \sim N(\mu, \tau^{-1})$$

$$\tau \sim \text{Gamma}(\alpha = 1, \beta = 36)$$

$$\mu|\tau \sim (\mu_0, (\theta\tau)^{-1})$$

  Since τ now follows a Gamma distribution, this allows the model to learn more from the data allowing for posterior mean to move closer to the data and the standard deviation to be wider compared to prior 1. Flexibility is now also increased due to this reason.

- Prior 3: The model assumes the same structure as prior 2, but with different hyperparameters/weaker prior. Due to the less informative priors, the posterior mean is almost entirely dependent on the data and the standard deviation is even wider. Flexibility is also higher than the previous two posteriors.

Due to the above observations, prior one is rigid and does not adapt to surprising information well. We can see that the credible region $(54.26549, 59.27997)$ does not cover any data in D2 as it is too high. Prior one also does not work very well for D1 for the same reason (61.7200366.73451). Additionally, the narrow width makes the region miss nearby/close data from D1.

Prior two is a lot more flexible than prior one as evident from the credible intervals, which are wider and captures the data better for both D1 and D2 groups.

Prior three is most flexible as credible intervals capture most of the data from D1 and D2 groups. Especially in the D2 group, the credible interval is much closer to the real data compared to the previous prior models.

Problem 3

(a)

Define

- X_i: Number of votes for purple party in district i
- θ_i: Probability that a voter in district i votes for purple party
- D_i: Indicator for whether purple party wins district i
- T: Indicator for whether the purple party wins the majority of districts

Then we can define our model:

$$X_i|\theta_i \sim Binomial(5001, \theta_i)$$

$$\theta_i \sim Beta(\alpha, \beta)$$

$$D_i = 1(X_i > 2500)$$

$$T = 1(\sum_{i=1}^{3} \geq 2)$$

(b)

For the non-informative prior, we can use the Uniform(0,1) prior

$$\theta_i \sim Beta(1, 1)$$

This prior assigns equal probability to all values of θ_i in the range $(0, 1)$, hence not favouring any particular voting probability, making it a non-informative prior.

For the informative prior, we can choose Beta(α, β) since the beta distribution is a conjugate prior for a binomial likelihood. We should choose α and β based on

- A mean of 0.5
- 95% of the probability mass falls within $(0.4, 0.6)$

The mean of Beta(α, β) is

$$\mathbb{E}[\theta] = \frac{\alpha}{\alpha + \beta}$$

So

$$\mathbb{E}[\theta] = \frac{\alpha}{\alpha + \beta} = 0.5 \implies \alpha = \beta$$

Now we need to solve for α such that $P(0.4 \leq \theta \leq 0.6) \approx 0.95$.

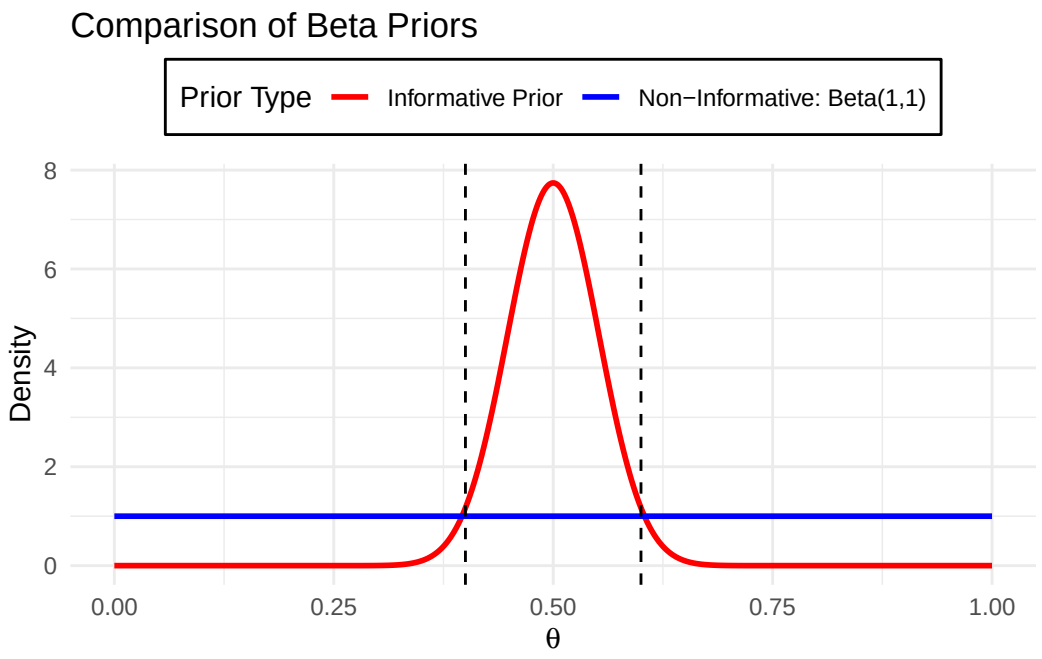
```
library(ggplot2)

mean_target <- 0.50
credible_interval <- c(0.40, 0.60)

# Function to calculate P(0.40 <= theta <= 0.6) = 0.95
check_credible_interval <- function(alpha) {
  beta <- alpha * (1 - mean_target) / mean_target
  prob_mass <- pbeta(credible_interval[2], alpha, beta) - pbeta(credible_interval[1], alpha,
  return(abs(prob_mass - 0.95))
}

best_alpha <- optimize(check_credible_interval, interval = c(1, 100))$minimum
best_beta <- best_alpha * (1 - mean_target) / mean_target
cat("Best Alpha:", best_alpha, "\n")
```

```
Best Alpha: 47.29981
```

```r
cat("Best Beta:", best_beta, "\n")
```

```
Best Beta: 47.29981
```

```r
theta_values <- seq(0, 1, length.out = 1000)
beta_data <- data.frame(
  theta = rep(theta_values, 2),
  density = c(dbeta(theta_values, 1, 1), dbeta(theta_values, best_alpha, best_beta)),
  prior_type = factor(rep(c("Non-Informative: Beta(1,1)", "Informative Prior"), each = 1000))
)

ggplot(beta_data, aes(x = theta, y = density, color = prior_type)) +
  geom_line(size = 1) +
  scale_color_manual(values = c("red", "blue")) +
  labs(title = "Comparison of Beta Priors", x = expression(theta), y = "Density", color = "Pr
  geom_vline(xintercept = credible_interval, linetype = "dashed", color = "black") +
  theme_minimal() +
  theme(legend.position = "top", legend.background = element_rect(fill = "white", color = "bl
```



(c)

```
library(ggplot2)
library(dplyr)
```

```
Attaching package: 'dplyr'
```

```
The following object is masked from 'package:MASS':

    select
```

```
The following objects are masked from 'package:stats':

    filter, lag
```

```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

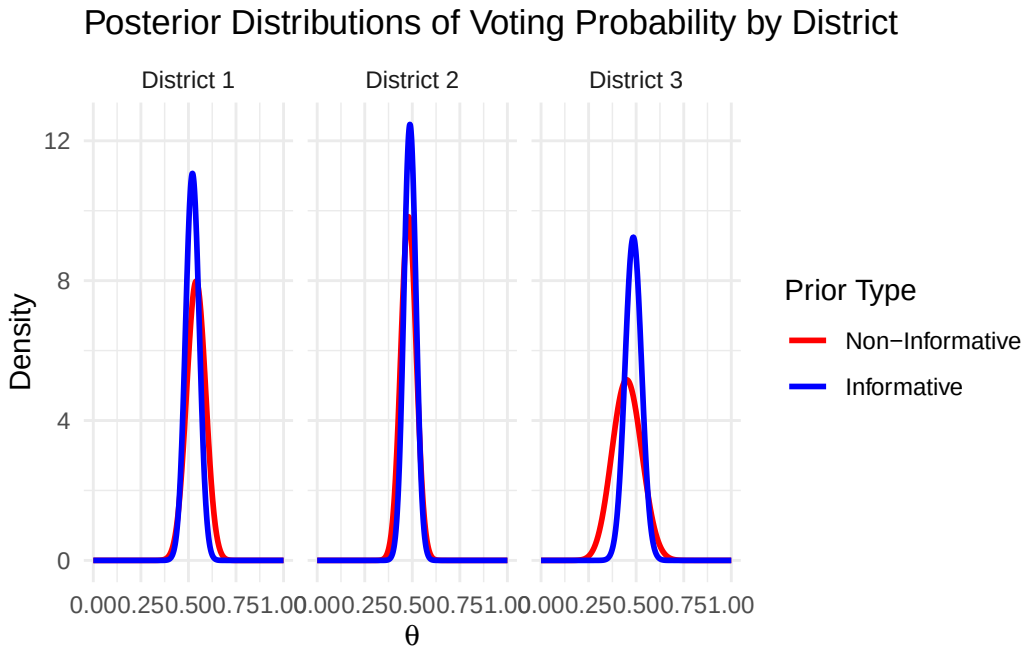
```
library(tidyr)

district_data <- data.frame(
  district = c("District 1", "District 2", "District 3"),
  votes_purple = c(53, 72, 18),
  total_votes = c(98, 150, 40)
)
priors <- list(
  "Non-Informative" = c(1, 1),
  "Informative" = c(best_alpha, best_beta)  # From Part (b)
)

# Posterior update
compute_posterior <- function(alpha_prior, beta_prior, votes_purple, total_votes) {
  alpha_post <- alpha_prior + votes_purple
  beta_post <- beta_prior + (total_votes - votes_purple)
  data.frame(
    posterior_mean = alpha_post / (alpha_post + beta_post),
    posterior_sd = sqrt((alpha_post * beta_post) / ((alpha_post + beta_post)^2 * (alpha_post
```

```
    ci_lower = qbeta(0.025, alpha_post, beta_post),
    ci_upper = qbeta(0.975, alpha_post, beta_post)
  )
}

# Compute posterior for each district and prior
posterior_results <- expand.grid(district = district_data$district, prior = names(priors)) %>
  rowwise() %>%
  mutate(
    votes_purple = district_data$votes_purple[district_data$district == district],
    total_votes = district_data$total_votes[district_data$district == district],
    alpha_prior = priors[[prior]][1],
    beta_prior = priors[[prior]][2],
    stats = list(compute_posterior(alpha_prior, beta_prior, votes_purple, total_votes))
  ) %>%
  unnest(stats)
print(posterior_results)
```

```
# A tibble: 6 x 10
  district  prior votes_purple total_votes alpha_prior beta_prior posterior_mean
  <fct>     <fct>        <dbl>       <dbl>       <dbl>      <dbl>          <dbl>
1 District~ Non-~           53          98         1          1            0.54
2 District~ Non-~           72         150         1          1            0.480
3 District~ Non-~           18          40         1          1            0.452
4 District~ Info~           53          98        47.3       47.3          0.521
5 District~ Info~           72         150        47.3       47.3          0.488
6 District~ Info~           18          40        47.3       47.3          0.485
# i 3 more variables: posterior_sd <dbl>, ci_lower <dbl>, ci_upper <dbl>
```

```
# Generate posterior distributions
theta_values <- seq(0, 1, length.out = 1000)
posterior_data <- expand.grid(district = district_data$district, prior = names(priors), theta
  rowwise() %>%
  mutate(
    votes_purple = district_data$votes_purple[district_data$district == district],
    total_votes = district_data$total_votes[district_data$district == district],
    alpha_prior = priors[[prior]][1],
    beta_prior = priors[[prior]][2],
    density = dbeta(theta, alpha_prior + votes_purple, beta_prior + total_votes - votes_purpl
  )
```

```r
ggplot(posterior_data, aes(x = theta, y = density, color = prior)) +
  geom_line(size = 1) +
  facet_wrap(~district) +
  scale_color_manual(values = c("red", "blue")) +
  labs(title = "Posterior Distributions of Voting Probability by District",
       x = expression(theta),
       y = "Density",
       color = "Prior Type") +
  theme_minimal()
```



(d)

```r
districts <- c("District 1", "District 2", "District 3")
X <- c(53, 72, 18)  # Purple
n <- c(98, 150, 40) # Total
majority_votes <- n / 2
n_sims <- 10000
priors <- list(
  "Non-Informative" = c(1, 1),
  "Informative" = c(best_alpha, best_beta)  # From Part (b)
```

```
)


sim_results <- data.frame()
for (prior_name in names(priors)) {
  alpha_prior <- priors[[prior_name]][1]
  beta_prior <- priors[[prior_name]][2]

  district_wins <- matrix(0, nrow = n_sims, ncol = length(districts))
  overall_wins <- numeric(n_sims)

  for (sim in 1:n_sims) {
    district_winners <- numeric(length(districts))

    for (i in 1:length(districts)) {
      theta_i <- rbeta(1, alpha_prior + X[i], beta_prior + (n[i] - X[i]))

      votes_purple <- rbinom(1, n[i], theta_i)
      district_winners[i] <- as.integer(votes_purple > majority_votes[i])
    }

    district_wins[sim, ] <- district_winners
    overall_wins[sim] <- as.integer(sum(district_winners) >= 2)
  }

  district_probs <- colMeans(district_wins)
  election_win_prob <- mean(overall_wins)

  temp <- data.frame(
    District = c(districts, "Overall Election"),
    Prior = prior_name,
    Win_Probability = c(district_probs, election_win_prob)
  )
  sim_results <- rbind(sim_results, temp)
}
print(sim_results)
```

```
          District           Prior Win_Probability
1       District 1 Non-Informative          0.6925
2       District 2 Non-Informative          0.3411
3       District 3 Non-Informative          0.2989
4 Overall Election Non-Informative          0.4027
```

```
5       District 1     Informative          0.6048
6       District 2     Informative          0.3882
7       District 3     Informative          0.3794
8 Overall Election     Informative          0.4312
```

Problem 4

(a)

Given that $\bar{Y_1} = -1.82, S_1 = 0.21$ We assume the likelihood

$$\bar{Y_1}|\theta, S_1 \sim N(\theta, S_1^2)$$

Since the prior is improper, the posterior distribution for θ follows the same form as the likelihood

$$\theta|\bar{Y_1}, S_1 \sim N(\bar{Y_1}, S_1^2) = N(-1.82, 0.21^2) = N(-1.82, 0.0441)$$

(b)

We use the information from the part (a) as prior and use the information in the new study with 79 patients as posterior. We can calculate the posterior mean

$$\mu' = \frac{\tau_0\mu_0 + n\tau\bar{x}}{\tau_0 + n\tau} = \frac{\mu_1/\sigma_1^2 + \mu_2/\sigma_2^2}{1/\sigma_1^2 + 1/\sigma_2^2} = \frac{\bar{Y_1}/S_1^2 + \bar{Y_2}/S_2^2}{1/S_1^2 + 1/S_2^2} = \frac{-1.82/0.21^2 + (-1.02)/0.28^2}{1/0.21^2 + 1/0.28^2} = -1.532$$

And calculate the posterior variance

$$\sigma'^2 = \frac{1}{1/S_1^2 + 1/S_2^2} = \frac{1}{1/0.21^2 + 1/0.28^2} = 0.028224$$

Thus, our new and updated posterior distribution is

$$\theta|\bar{Y_1}, \bar{Y_2}, S_1, S_2 \sim N(-1.532, \sqrt{0.0282})$$

(c)

From the new study, we have

$$\bar{Y_2} = -1.02, \quad S_2 = 0.28$$

Similar to previous part, the posterior distribution will be

$$\theta|\bar{Y_2}, S_2 \sim N(-1.02, 0.28^2) = N(-1.02, 0.0784)$$

Now my colleague learns about my study and updates her beliefs based on my data. Recall that my trial had data

$$\bar{Y_1} = -1.82, S_1 = 0.21$$

So using the same formula to part (b), we can calculate that

$$\mu' = -1.532, \quad \sigma'^2 = 0.028224$$

Hence, her new and updated posterior distribution is

$$\theta|\bar{Y_1}, \bar{Y_2}, S_1, S_2 \sim N(-1.532, \sqrt{0.0282})$$

(d)

Given these 7 studies, let us set $\tau_0 = 0$. Then we can simplify the expressions:

$$\frac{\tau_0\mu_0 + \sum_i \tau_i Y_i}{\tau_0 + \sum_i \tau_i} = \frac{\sum_i \tau_i Y_i}{\sum_i \tau_i}$$

$$\tau_0 + \sum_i \tau = \sum_i \tau_i$$

We can find the posterior distribution of θ using the following:

$$\theta \sim N\left(\frac{\sum_i \tau_i Y_i}{\sum_i \tau_i}, \sqrt{\frac{1}{\sum_i \tau_i}}\right)$$

For each trial, we have:

- **Trial 1:** $\tau_1 = \frac{1}{0.21^2} = \frac{1}{0.0441}$
- **Trial 2:** $\tau_2 = \frac{1}{0.28^2} = \frac{1}{0.0784}$
- **Trial 3:** $\tau_3 = \frac{1}{0.945^2} = 0.893025$
- **Trial 4:** $\tau_4 = \frac{1}{0.285^2} = 0.081225$

- **Trial 5:** $\tau_5 = \frac{1}{0.545^2} = 0.297025$

From which we can calculate:

$$\sum_i \tau_i Y_i = \frac{1}{0.0441} \times (-1.82) + \frac{1}{0.0784} \times (-1.02) + \frac{1}{0.893025} \times (-1.90) + \frac{1}{0.081225} \times (-2.00) + \frac{1}{0.297025} \times (-1.21) = -$$

$$\sum_i \tau_i = \frac{1}{0.0441} + \frac{1}{0.0784} + \frac{1}{0.893025} + \frac{1}{0.081225} + \frac{1}{0.297025} = 52.22882891$$

Finally, we have posterior mean and variance

$$\mu' = \frac{-85.10433744}{52.22882891} = -1.629451382$$

$$\sigma'^2 = \frac{1}{\sum_i \tau_i} = \frac{1}{52.22882891}$$

Thus, our posterior distribution for θ is:

$$\theta \sim N(-1.629451382, \sqrt{52.22882891})$$