

# \* Assignment - 2 \*

Q.1.

```
1.1 # Number of successes
y <- 7

# Prior distribution
prior <- function(theta) {
  if (theta >= 0 && theta <= 1) {
    return(1)
  } else {
    return(0)
  }
}

# Likelihood function
likelihood <- function(theta) {
  return(choose(10, y) * theta^y * (1 - theta)^(10 - y))
}

# Posterior density function
posterior <- function(theta) {
  marginal_likelihood <- 1/11
  posterior_density <- (likelihood(theta) * prior(theta)) / marginal_likelihood
  return(posterior_density)
}

# Values of theta to estimate the posterior density
theta_values <- c(0.75, 0.25, 1)

# Calculate the posterior densities for the given theta values
posterior_densities <- sapply(theta_values, function(theta) posterior(theta))
results <- data.frame(theta = theta_values, posterior_density = posterior_densities)
print(results)
```

1.2

```
theta_values <- seq(0, 1, length.out = 1000)

# Calculate the posterior densities for the given theta values
posterior_densities <- sapply(theta_values, function(theta) posterior(theta))
results <- data.frame(theta = theta_values, posterior_density = posterior_densities)

# Plot of posterior density
plot(results$theta, results$posterior_density, type = "l", col = "blue", lwd = 2,
      xlab = expression(theta), ylab = "Posterior Density", main = "Posterior Density of
Theta")
```

1.3

```
max_posterior_theta <- theta_values[which.max(posterior_densities)]
cat("theta for maximum posterior", max_posterior_theta)
likelihood_values <- sapply(theta_values, function(theta) likelihood(theta))
prior_values <- sapply(theta_values, prior)
posterior_densities <- sapply(theta_values, function(theta) posterior(theta))
```

1.4

```
# graphs of likelihood function, prior distribution, posterior distribution
plot(theta_values, likelihood_values, type = "l", col = "red", lwd = 2,
      main = "Likelihood, Prior, and Posterior Distributions",
      xlab = "", ylab = "Density", ylim = c(0, max(posterior_densities)))
lines(theta_values, prior_values, col = "green", lwd = 2)
lines(theta_values, posterior_densities, col = "blue", lwd = 2)
legend("topright", legend = c("Likelihood", "Prior", "Posterior"),
      col = c("red", "green", "blue"), lwd = 2)
```

1.1.

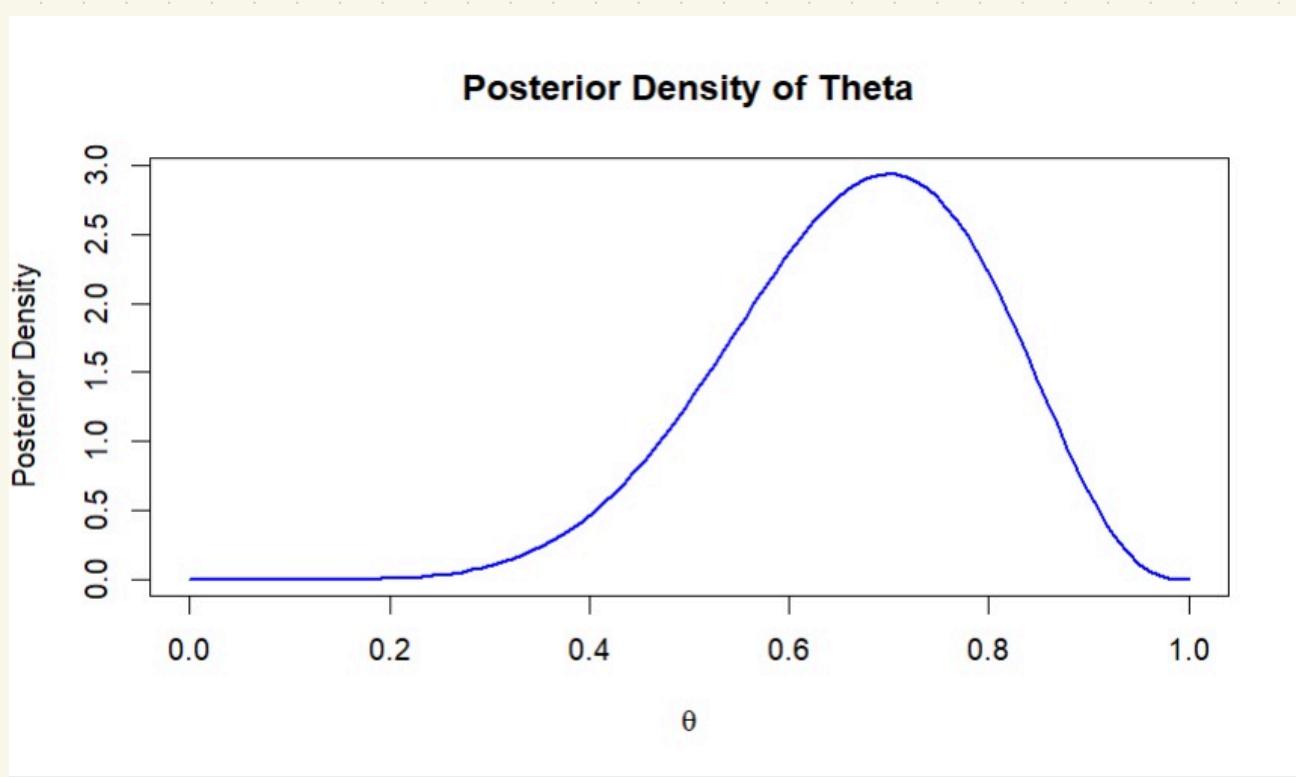
theta posterior density

0.75	2.75310516
0.25	0.03398895
1.00	0.00000000

1.3.

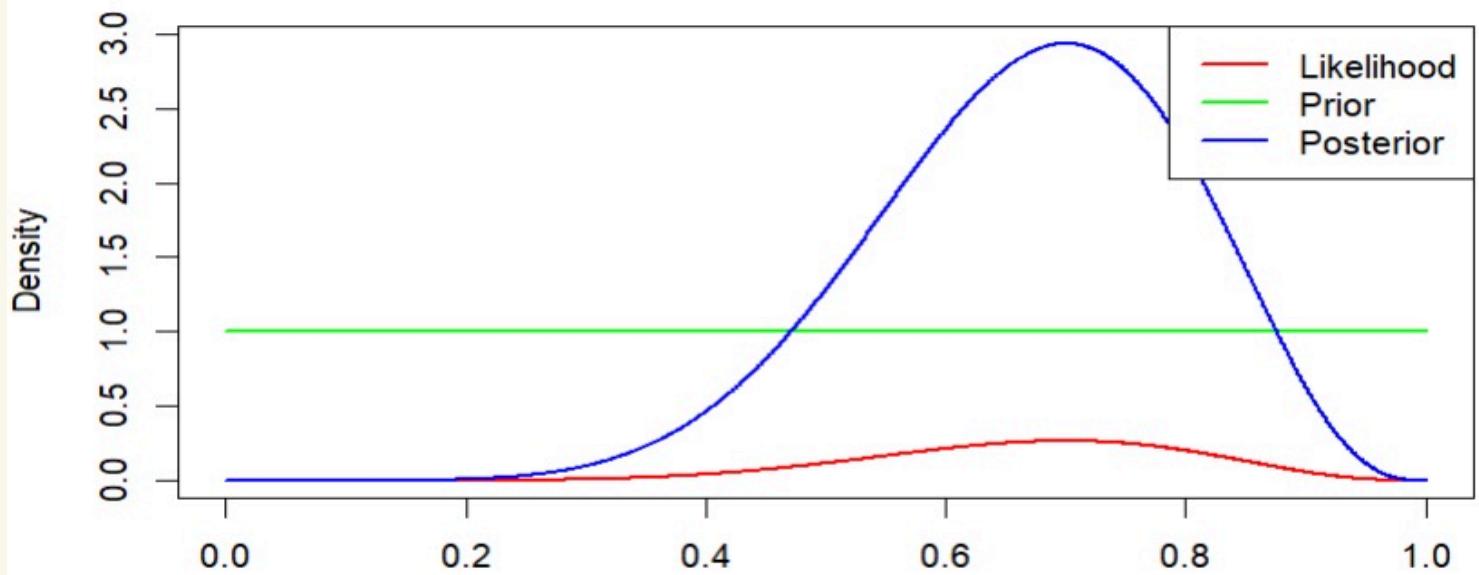
theta for maximum posterior 0.6996997

1.2.



1.4

### Likelihood, Prior, and Posterior Distributions



By these plots we can conclude that likelihood and posterior contain same quantity of information because we have a constant prior ( $= 1 \text{ for } \theta \in [0,1]$ )

Q.2.

```
# Define the data
y <- c(300, 270, 390, 450, 500, 290, 680, 450)
n <- length(y)
sigma <- 50

# Define the likelihood function
likelihood <- function(mu) {
  return((1 / (sigma * sqrt(2 * pi)))^n * exp(-1 / (2 * sigma^2) * sum((y - mu)^2)))
}

# Define the prior function
prior <- function(mu) {
  mu_mean <- 250
  mu_sd <- 25
  return((1 / (sqrt(2 * pi) * mu_sd)) * exp(-1 / (2 * mu_sd^2) * (mu - mu_mean)^2))
}

# Define the unnormalized posterior function
unnormalized_posterior <- function(mu) {
  return(likelihood(mu) * prior(mu))
}
```

# Calculate the unnormalized posterior for given mu values
mu\_values <- c(300, 900, 50)
posterior\_values <- sapply(mu\_values, unnormalized\_posterior)
results <- data.frame(mu = mu\_values, unnormalized\_posterior = posterior\_values)
print(results)

# Define a sequence of mu values for plotting
mu\_seq <- seq(100, 500, length.out = 1000)

# Calculate the unnormalized posterior for the sequence of mu values
posterior\_seq <- sapply(mu\_seq, unnormalized\_posterior)

# Plot the unnormalized posterior distribution
plot(mu\_seq, posterior\_seq, type = "l", col = "blue", lwd = 2,
 xlab = expression(mu), ylab = "Unnormalized Posterior Density",
 main = "Unnormalized Posterior Distribution of Mu")

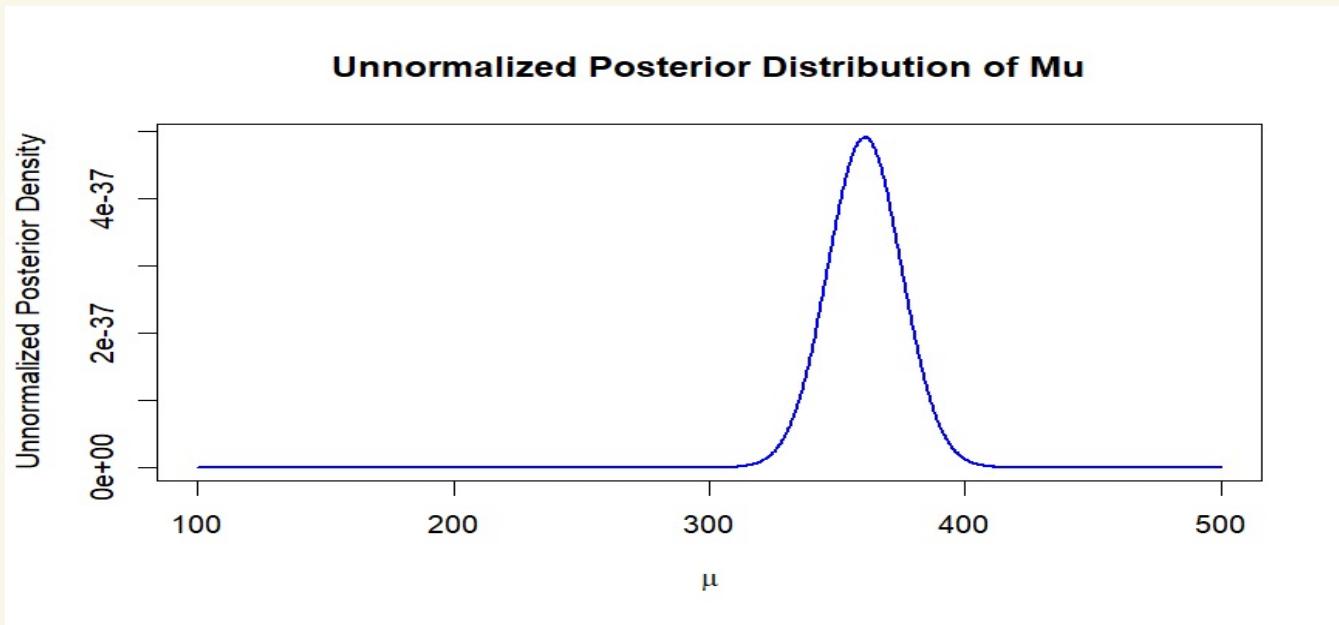
# Calculate the prior and unnormalized posterior for the sequence of mu values
prior\_seq <- sapply(mu\_seq, prior)

# Plot the prior distribution
plot(mu\_seq, prior\_seq, type = "l", col = "green", lwd = 2,
 xlab = expression(mu), ylab = "Prior Density",
 main = "Prior Distribution of Mu")

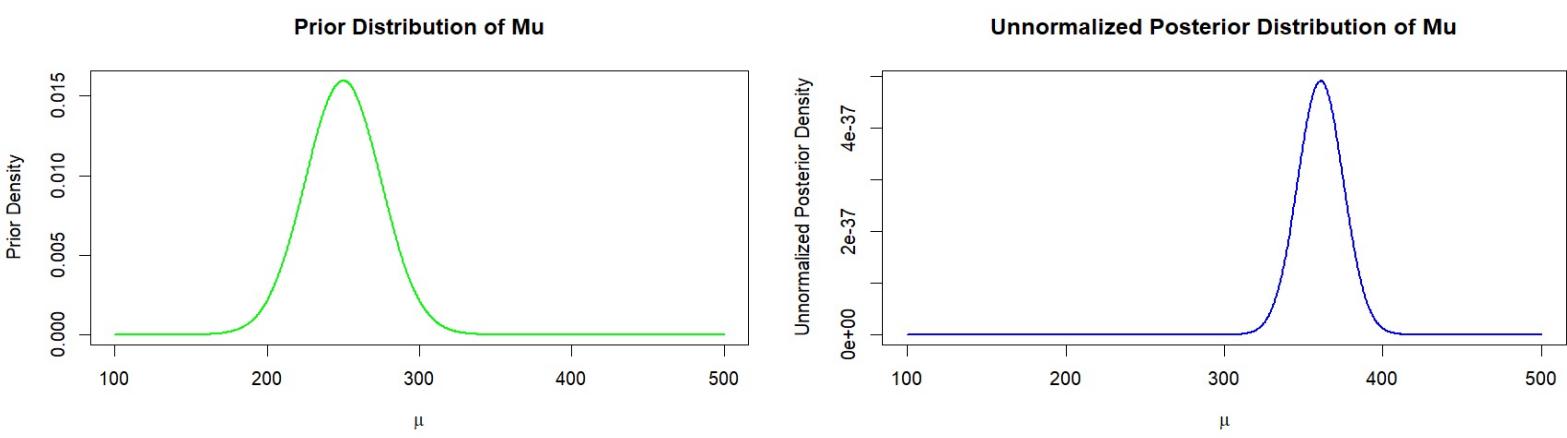
2.1.

mu	unnormalized_posterior
300	6.824248e-41
900	0.000000e+00
50	9.691374e-138

2.2.



2.3.



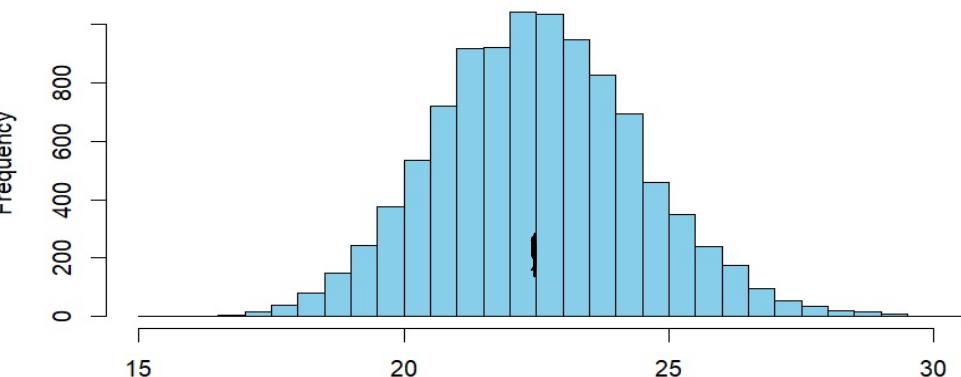
The  $\max^m$  of posterior ( $w.r.t \mu$ ) is  $> 250$  which is quite expected as the observations given have avg  $> 250$ .

- 3.1 for day 1  $\rightarrow$  prior  $\sim$  Gamma (40, 2)  
 $(\lambda)$   
 posterior  $\sim$  Gamma (40 + 25, 3)  
 $(\lambda)$   
 $= \text{Gamma}(65, 3)$
- for day 2  $\rightarrow$  prior  $\sim$  Gamma (65, 3)  
 $(\lambda)$   
 posterior  $\sim$  Gamma (65 + 20, 4)  
 $(\lambda)$   
 $= \text{Gamma}(85, 4)$
- for day 3  $\rightarrow$  prior  $\sim$  Gamma (85, 4)  
 $(\lambda)$   
 posterior  $\sim$  Gamma (85 + 23, 5)  
 $(\lambda)$   
 $= \text{Gamma}(108, 5)$
- for day 4  $\rightarrow$  prior  $\sim$  Gamma (108, 5)  
 $(\lambda)$   
 posterior  $\sim$  Gamma (108 + 27, 6)  
 $(\lambda)$   
 $= \text{Gamma}(135, 6)$
- $\Rightarrow$  prior for day 5  $\sim$  Gamma (135, 6)  $= \underline{\text{Ans}}$

3.2:

```
posterior_sample_day5 <- rgamma(10000, shape = 135, rate = 6)
hist(posterior_sample_day5, breaks = 30, col = "skyblue",
  main = "Predicted Number of Road Accidents on Day 5",
  xlab = "Number of Road Accidents", ylab = "Frequency")
```

Predicted Number of Road Accidents on Day 5



Clearly by histogram  
 we can see no. of  
 accidents is between  
 20 & 25.

Q.4.5.1

```
# Load necessary libraries
library(truncnorm)

# Load the data from the given URL

dat <- read.table(
  "https://raw.githubusercontent.com/yadavhimanshu059/CGS698C/main/notes/Module-
2/recognition.csv", sep = ",", header = T) [, -1]
head(dat)

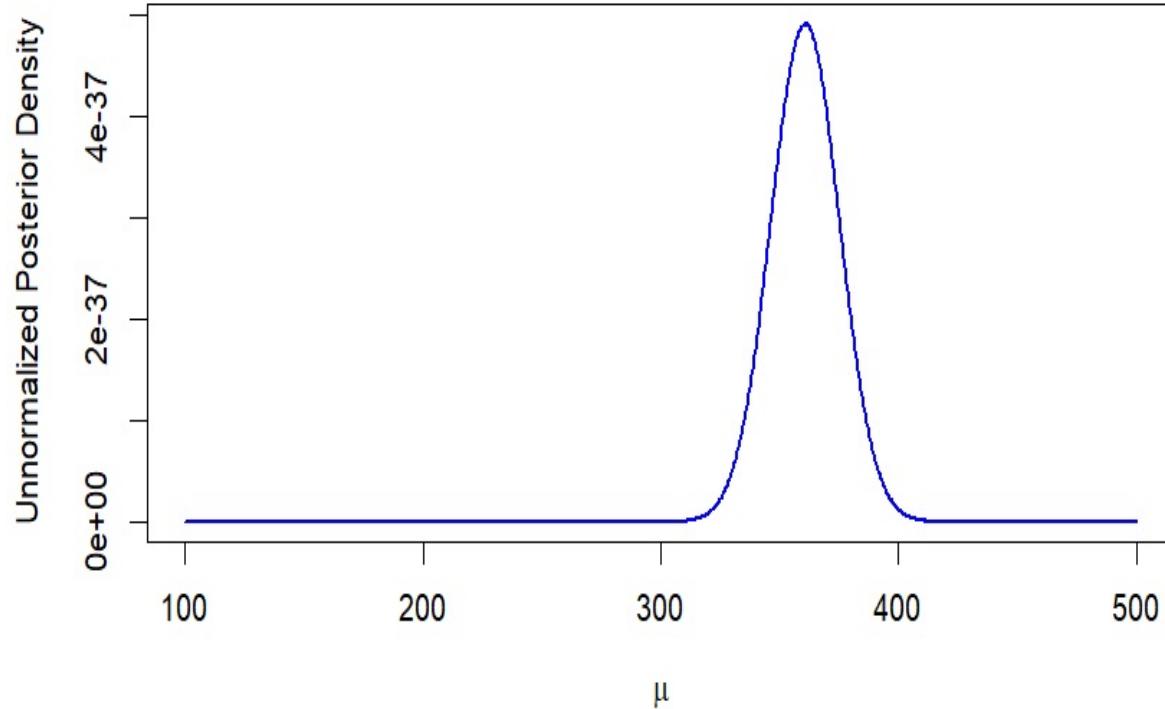
Tw <- dat$Tw
Tnw <- dat$Tnw

# Define the function to calculate the unnormalized posterior for the Null hypothesis
unnormalized_posterior <- function(mu, delta, Tw, Tnw, sigma) {
  L_Tw <- prod(dnorm(Tw, mean = mu, sd = sigma))
  L_Tnw <- prod(dnorm(Tnw, mean = mu + delta, sd = sigma))
  p_mu <- dnorm(mu, mean = 300, sd = 50)
  p_delta <- dtruncnorm(delta, a = 0, b = Inf, mean = 0, sd = 50)
  return(L_Tw * L_Tnw * p_mu * p_delta)
}
sigma <- 60
mu_values <- seq(200, 400, length.out = 100)
delta <- 0

# Compute the unnormalized posterior for each value of mu with delta fixed at 0
posterior_values <- sapply(mu_values, function(mu) unnormalized_posterior(mu, delta, Tw,
Tnw, sigma))

# Plot the unnormalized marginal posterior for mu
plot(mu_values, posterior_values, type = 'l', xlab = expression(mu), ylab = 'Unnormalized
Posterior Density', main = 'Unnormalized Posterior Distribution of Mu under Null
Hypothesis')
```

### Unnormalized Posterior Distribution of Mu



Q.4.5.2

```
# Load necessary library
library(truncnorm)

sigma <- 60

mu_samples <- rnorm(1000, mean = 300, sd = 50)
delta_samples <- rtruncnorm(1000, a = 0, b = Inf, mean = 0, sd = 50)

tw <- rnorm(1000, mean = mu_samples, sd = sigma)

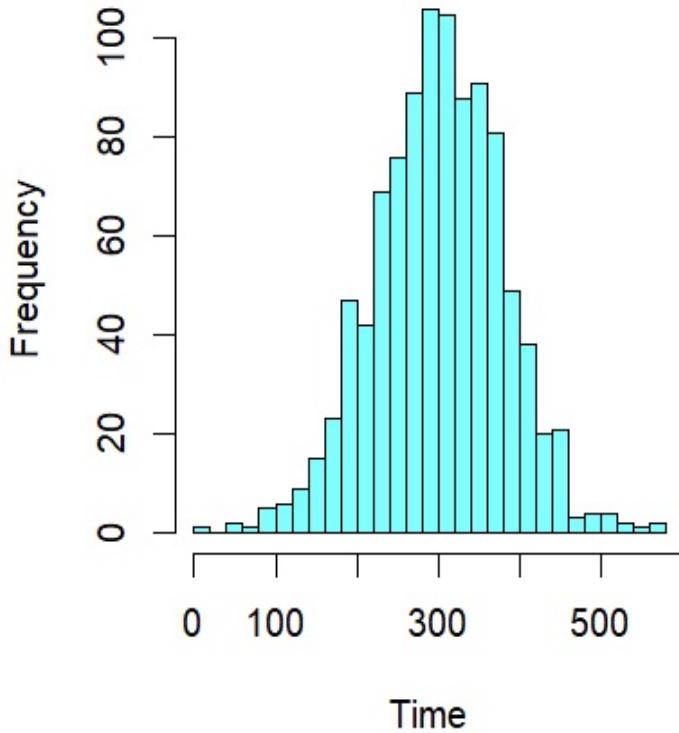
# Generate non-word recognition times from N(mu + delta, sigma = 60)
tnw <- rnorm(1000, mean = mu_samples + delta_samples, sd = sigma)

par(mfrow = c(1, 2))

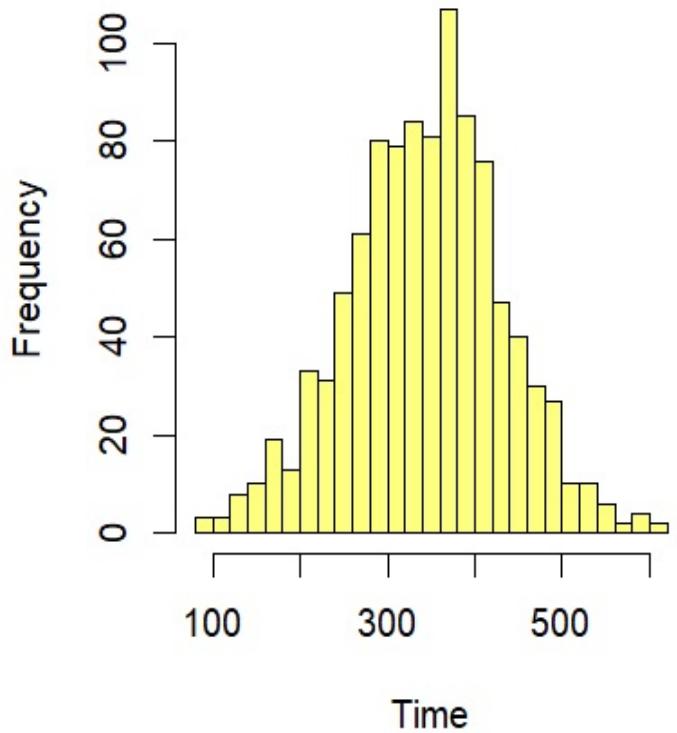
# Plot word recognition times
hist(tw, breaks = 30, col = rgb(0, 1, 1, 0.5), main = "Word Recognition Times", xlab =
"Time")

# Plot non-word recognition times
hist(tnw, breaks = 30, col = rgb(1, 1, 0, 0.5), main = "Non-word Recognition Times", xlab =
"Time")
```

**Word Recognition Times**



**Non-word Recognition Times**



Q. 4.5.3 .

```
# Load necessary library
library(truncnorm)

sigma <- 60

# Draw samples for mu from its prior distribution N(300, 50)
mu_samples <- rnorm(1000, mean = 300, sd = 300)

# Generate data for Null Hypothesis Model (delta = 0)
delta_null <- 0
tw_null <- rnorm(1000, mean = mu_samples, sd = sigma)
tnw_null <- rnorm(1000, mean = mu_samples + delta_null, sd = sigma)

# Generate data for Lexical Access Model (delta varies)
delta_samples <- rtruncnorm(1000, a = 0, b = Inf, mean = 0, sd = 50)
tw_lexical <- rnorm(1000, mean = mu_samples, sd = sigma)
tnw_lexical <- rnorm(1000, mean = mu_samples + delta_samples, sd = sigma)

# Plot histograms for Null Hypothesis Model
hist(tw_null, breaks = 30, col = rgb(0, 0, 1, 0.5), xlim = range(c(tw_null, tnw_null)),
main = "Histogram of tw_NULL", xlab = "Time")
hist(tnw_null, breaks = 30, col = rgb(1, 1, 0, 0.5), add = FALSE)

# Plot histograms for Lexical Access Model
hist(tw_lexical, breaks = 30, col = rgb(0, 0, 1, 0.5), xlim = range(c(tw_lexical,
tnw_lexical)), main = "Histogram of tw_lexical", xlab = "Time")
hist(tnw_lexical, breaks = 30, col = rgb(1, 1, 0, 0.5), add = FALSE) # Load necessary
library
library(truncnorm)

sigma <- 60

mu_samples <- rnorm(1000, mean = 300, sd = 50)
delta_samples <- rtruncnorm(1000, a = 0, b = Inf, mean = 0, sd = 50)
tw <- rnorm(1000, mean = mu_samples, sd = sigma)

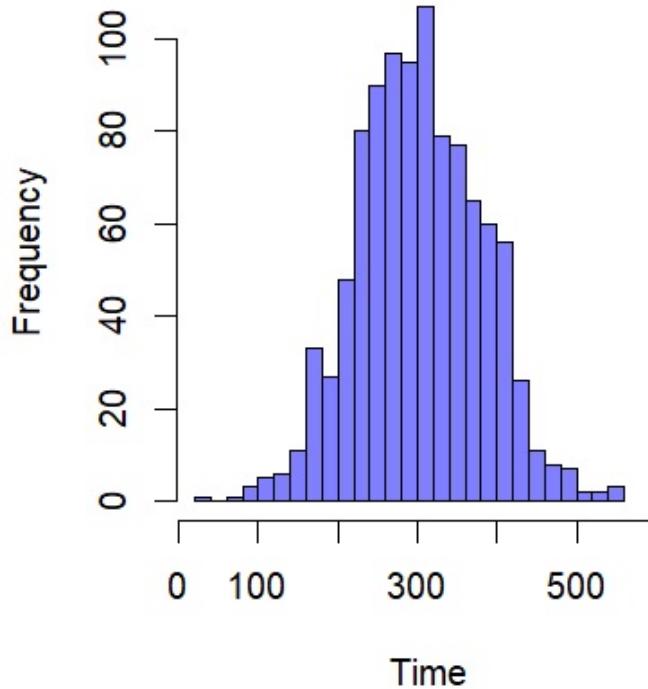
# Generate non-word recognition times from N(mu + delta, sigma = 60)
tnw <- rnorm(1000, mean = mu_samples + delta_samples, sd = sigma)

par(mfrow = c(1, 2))

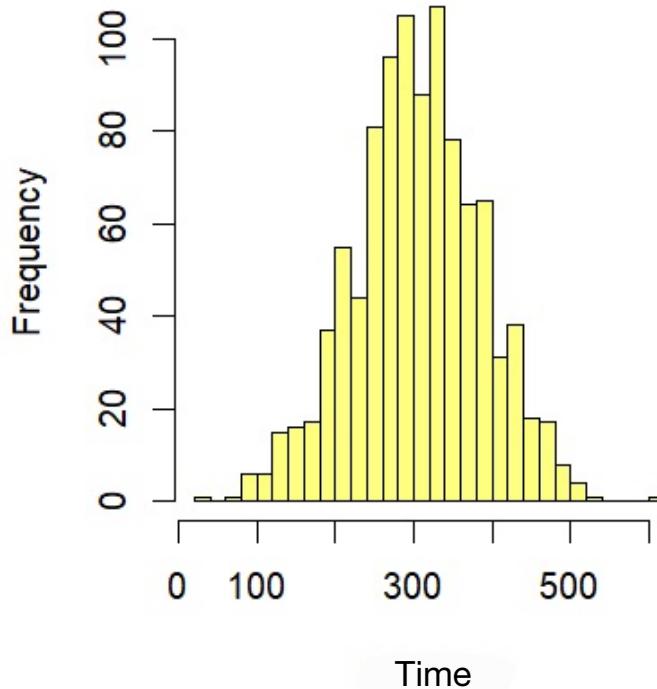
# Plot word recognition times
hist(tw, breaks = 30, col = rgb(0, 1, 1, 0.5), main = "Word Recognition Times", xlab =
"Time")

# Plot non-word recognition times
hist(tnw, breaks = 30, col = rgb(1, 1, 0, 0.5), main = "Non-word Recognition Times", xlab =
"Time")
```

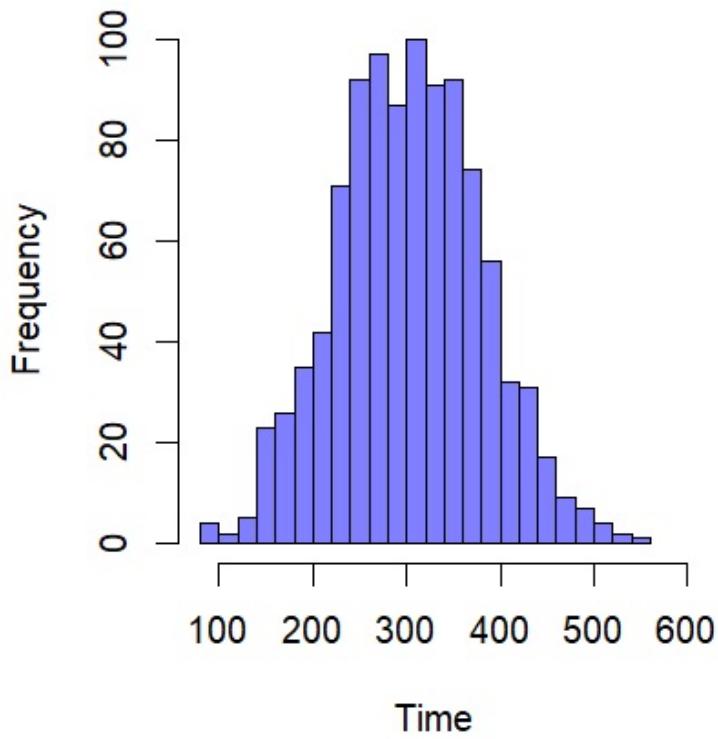
### Histogram of tw\_NULL



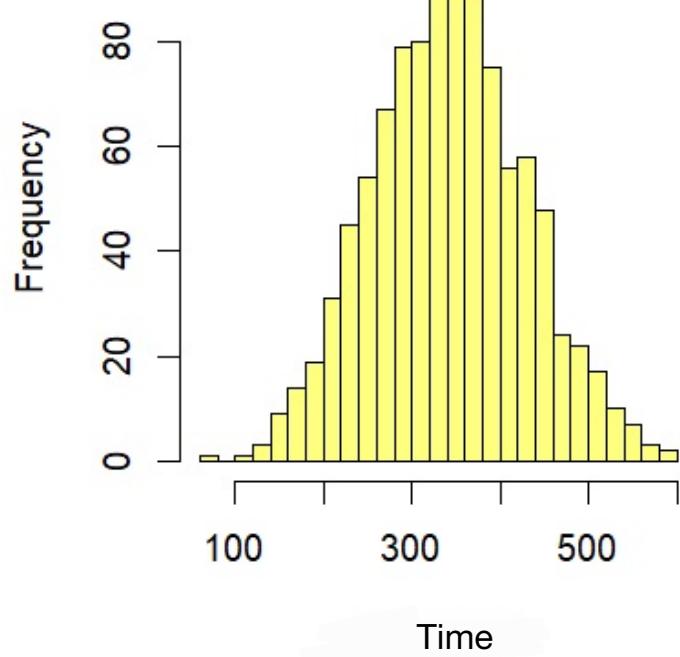
### Histogram of tnw\_null



### Histogram of tw\_lexical



### Histogram of tnw\_lexical



Q.M.S.M

```
# Set seed for reproducibility
set.seed(123)

# Generate prior samples for mu and delta for both models
mu_values <- runif(50000, min = 100, max = 500)
delta_values <- runif(50000, min = 0, max = 50)

# Set common parameters
sigma <- 60

# Generate predicted values for Tw and Tnw based on the priors for both models
predicted_Tw_model1 <- rnorm(length(dat$Tw), mean = mean(mu_values), sd = sigma)
predicted_Tnw_model1 <- rnorm(length(dat$Tnw), mean = mean(mu_values), sd = sigma)

predicted_Tw_model2 <- rnorm(length(dat$Tw), mean = mean(mu_values), sd = sigma)
predicted_Tnw_model2 <- rnorm(length(dat$Tnw), mean = mean(mu_values + delta_values), sd = sigma)

# Plot histograms of the observed data and the predicted values
par(mfrow = c(2, 2))

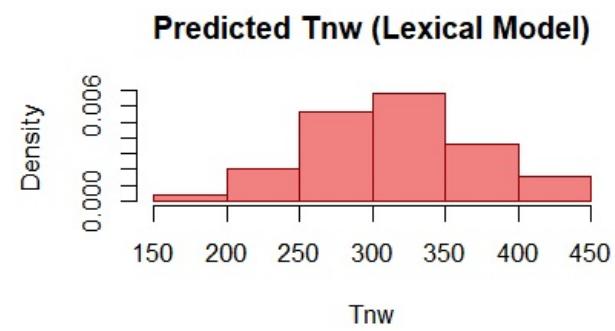
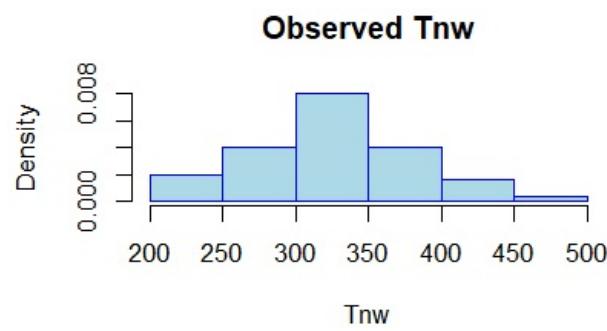
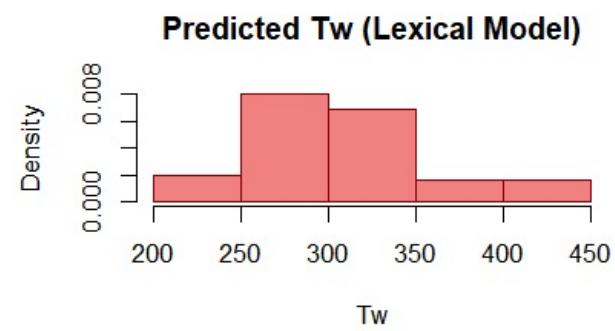
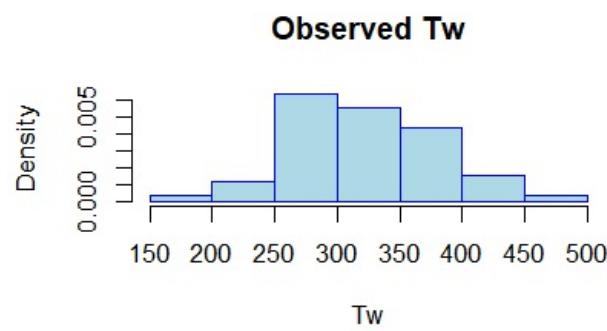
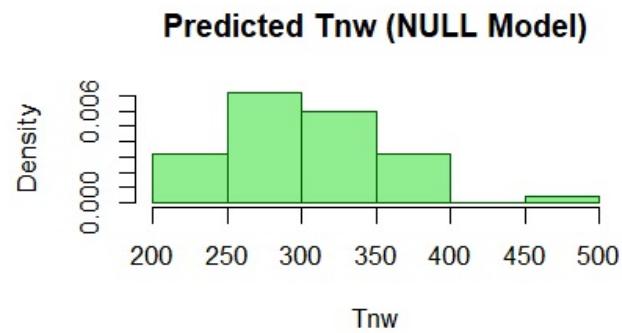
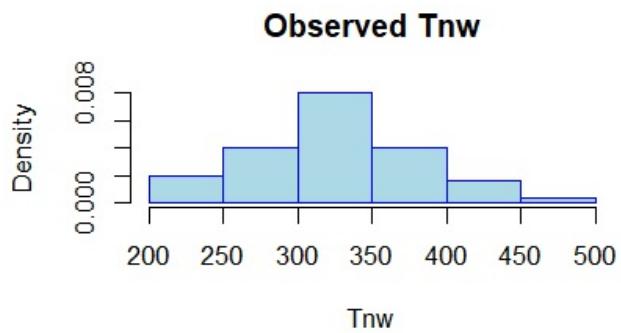
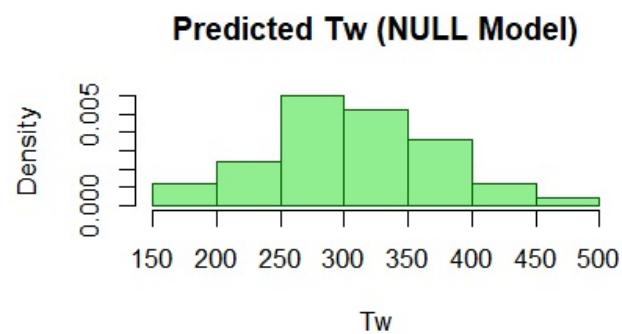
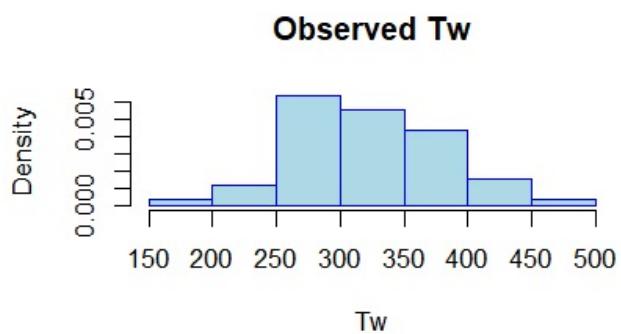
# Histograms for NULL Model
hist(dat$Tw, main = "Observed Tw", xlab = "Tw", freq = FALSE, col = "lightblue", border = "blue")
hist(predicted_Tw_model1, main = "Predicted Tw (NULL Model)", xlab = "Tw", freq = FALSE, col = "lightgreen", border = "darkgreen")

hist(dat$Tnw, main = "Observed Tnw", xlab = "Tnw", freq = FALSE, col = "lightblue", border = "blue")
hist(predicted_Tnw_model1, main = "Predicted Tnw (NULL Model)", xlab = "Tnw", freq = FALSE, col = "lightgreen", border = "darkgreen")

# Histograms for Lexical Model
hist(dat$Tw, main = "Observed Tw", xlab = "Tw", freq = FALSE, col = "lightblue", border = "blue")
hist(predicted_Tw_model2, main = "Predicted Tw (Lexical Model)", xlab = "Tw", freq = FALSE, col = "lightcoral", border = "darkred")

hist(dat$Tnw, main = "Observed Tnw", xlab = "Tnw", freq = FALSE, col = "lightblue", border = "blue")
hist(predicted_Tnw_model2, main = "Predicted Tnw (Lexical Model)", xlab = "Tnw", freq = FALSE, col = "lightcoral", border = "darkred")

par(mfrow = c(1, 1))
```



Clearly from plots the overlapping Content of Tw & Tnw with observed data is more in NULL hypothesis model.

Hence

\*NULL hypothesis\* model is more consistent with data

```

# Load necessary library
library(truncnorm)

# Load the data from the given URL
url <- "https://raw.githubusercontent.com/yadavhimanshu059/CGS698C/main/notes/Module-2/recognition.csv"
dat <- read.csv(url)

# Extract Tw and Tnw from the data
Tw <- dat$Tw
Tnw <- dat$Tnw

# Set parameters
mu_mean <- 300
mu_sd <- 50
sigma <- 60
n_points <- 1000 # Number of points to evaluate

# Define the likelihood function
likelihood <- function(mu, delta, Tw, Tnw, sigma) {
  L_Tw <- prod(dnorm(Tw, mean = mu, sd = sigma))
  L_Tnw <- prod(dnorm(Tnw, mean = mu + delta, sd = sigma))
  return(L_Tw * L_Tnw)
}

# Define the prior for mu and delta
p_mu <- function(mu) {
  return(dnorm(mu, mean = mu_mean, sd = mu_sd))
}

p_delta <- function(delta) {
  return(dtruncnorm(delta, a = 0, b = Inf, mean = 0, sd = 50))
}

# Define the unnormalized posterior function for delta
unnormalized_posterior_delta <- function(delta, mu_values, Tw, Tnw, sigma) {
  likelihoods <- sapply(mu_values, function(mu) likelihood(mu, delta, Tw, Tnw, sigma))
  return(sum(likelihoods) * p_delta(delta))
}

# Generate samples for mu from its prior distribution N(300, 50)
mu_values <- rnorm(1000, mean = mu_mean, sd = mu_sd)

# Generate a sequence of delta values
delta_values <- rtruncnorm(1000, a=0, b=Inf, mean=0, sd=50)

# Compute the unnormalized posterior distribution for delta
posterior_delta_values <- sapply(delta_values, function(delta)
  unnormalized_posterior_delta(delta, mu_values, Tw, Tnw, sigma))

set.seed(123) # For reproducibility
samples <- sample(delta_values, size = 1000, replace = TRUE, prob =
posterior_delta_values)

# Plot the histogram of the sampled delta values
hist(samples, breaks = 30, col = "lightblue", border = "blue",
      main = "Histogram of Unnormalized Posterior Distribution of  $\delta$ ",
      xlab = " $\delta$ ", ylab = "Frequency")

```

### Histogram of Unnormalized Posterior Distribution of $\delta$

