CGS698
Assignment-2

Anupam Chaudhary
(210170)

Part-1:A Simple Binomial Model

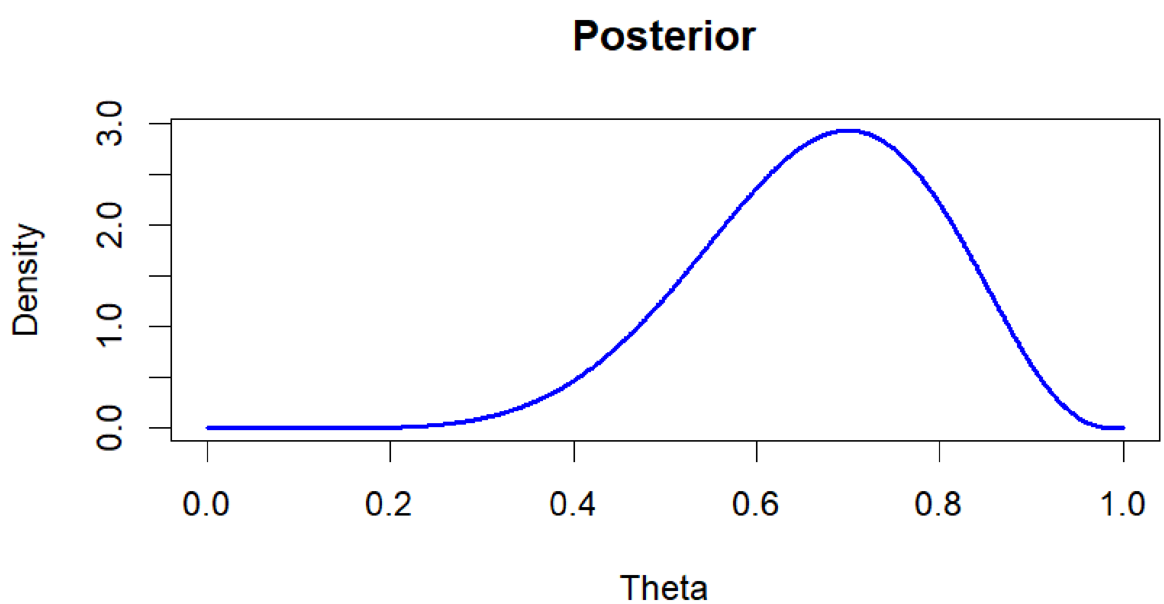
1.1)

a) θ=0.75, Posterior = `2.753105`

b) θ=0.25, Posterior = `0.03398895`

c) θ=1, Posterior = `0`

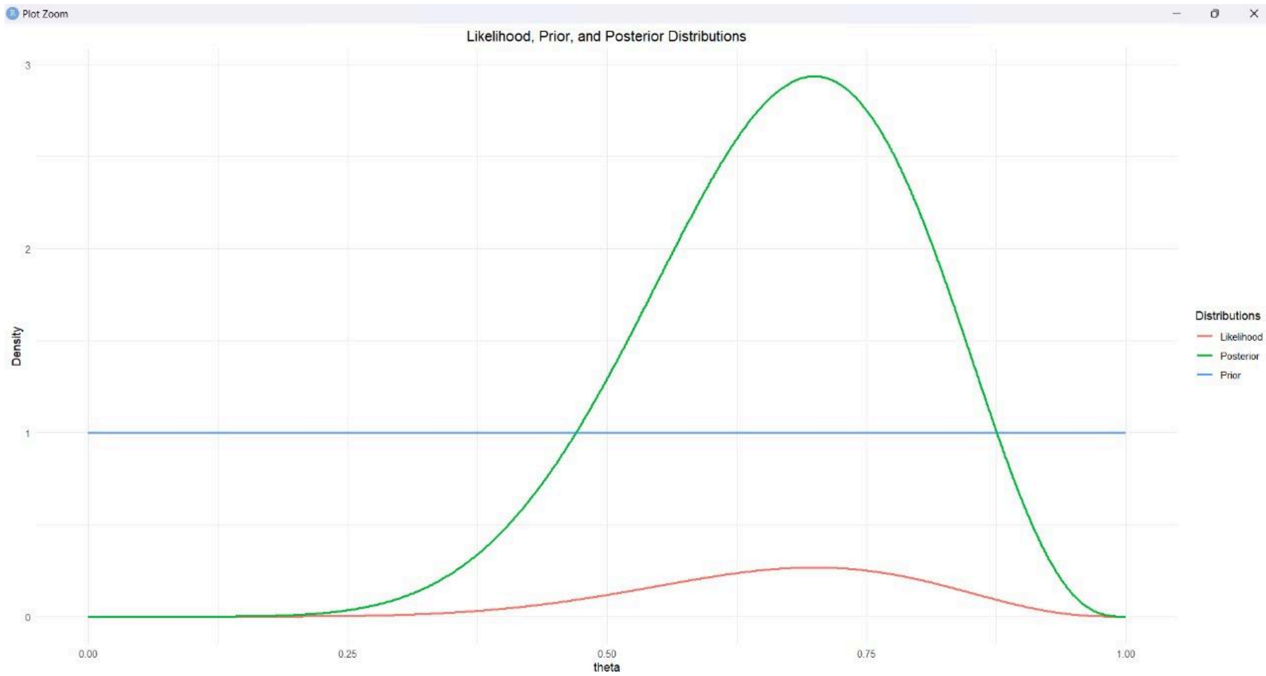
1.2)



1.3)

Value of θ has the maximum posterior density = `0.6996997`.Also we can see this from graph below.

1.4)



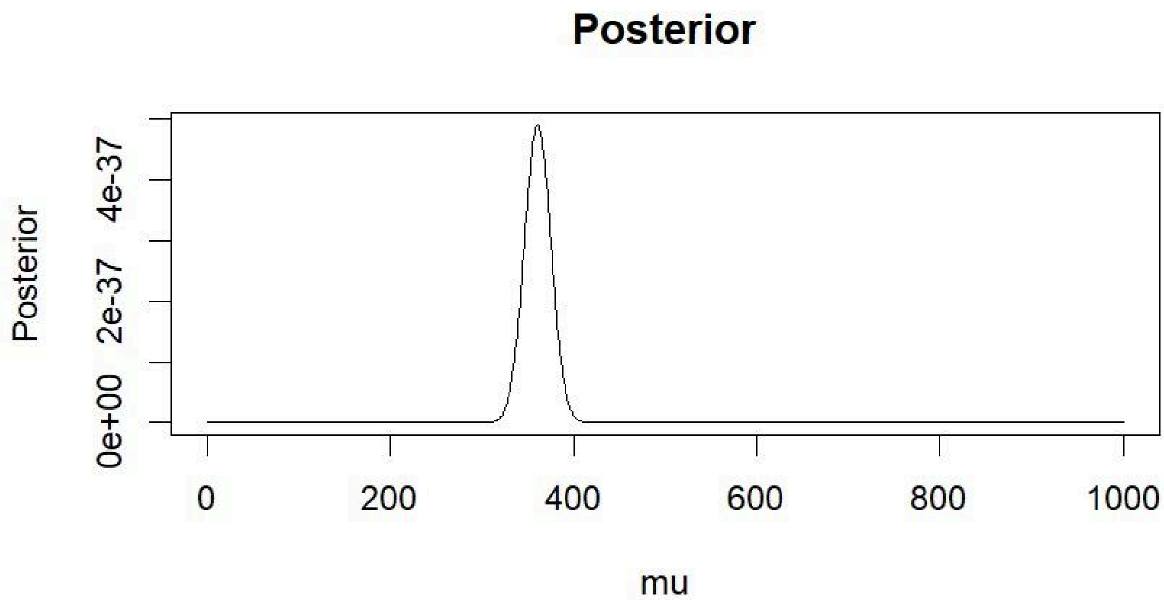
Part-2:A Gaussian Model of Reading

2.1)

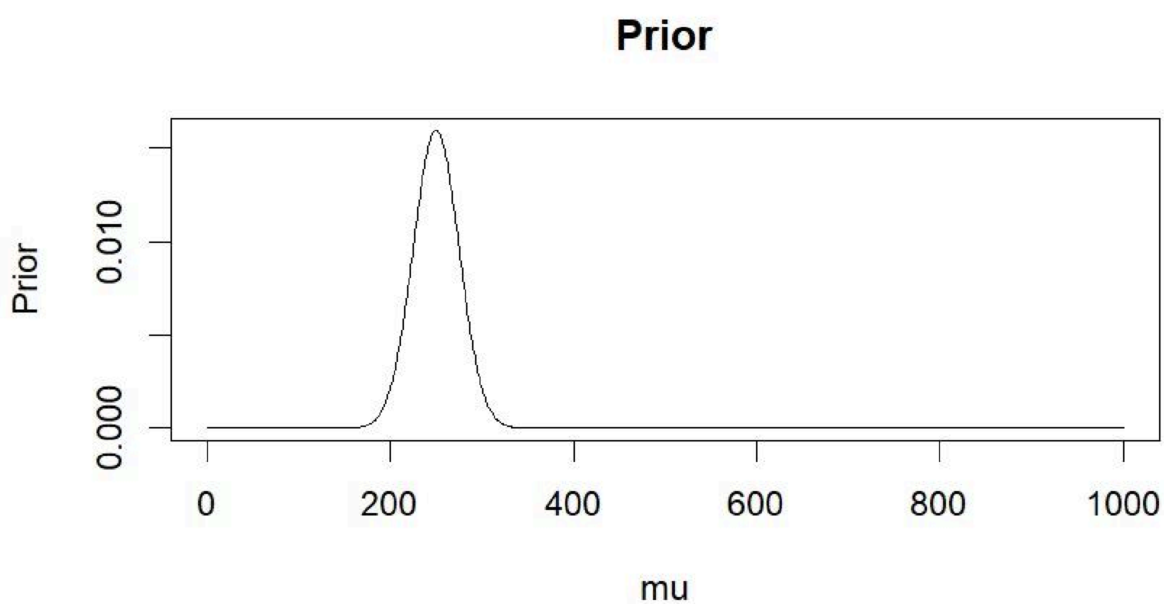
The unnormalized posterior density for the following values of μ:

- $\mu = 300$
  unnormalized posterior density = 6.824248×10^{-41}
- $\mu = 900$
  unnormalized posterior density = nearly 0
- $\mu = 50$
  unnormalized posterior density = $9.691374 \times 10^{-138}$

2.2)


Posterior
Posterior
0e+00
2e-37
4e-37
0
200
400
600
800
1000
mu

2.3)


Prior
Prior
0.000
0.010
0
200
400
600
800
1000
mu

Part-3:The Bayesian Learning

3.1)

$$\lambda \sim \text{Gamma}(40+k,3)$$

To generate predictions for Day 5, we use the posterior distribution obtained after Day 4 as the prior distribution for Day 5. Similarly, the prior distribution of any day depends on its previous day posterior distribution.

For,**Day-1**

$$\text{Prior} = \lambda \sim \text{Gamma}(40,2)$$
$$\text{Posterior} = \lambda \sim \text{Gamma}(40+25,2+1) = \text{Gamma}(65,3)$$

For,**Day-2**

$$\text{Prior} = \lambda \sim \text{Gamma}(65,3)$$
$$\text{Posterior} = \lambda \sim \text{Gamma}(65+20,3+1) = \text{Gamma}(85,4)$$

For,**Day-3**

$$\text{Prior} = \lambda \sim \text{Gamma}(85,4)$$
$$\text{Posterior} = \lambda \sim \text{Gamma}(85+23,4+1) = \text{Gamma}(108,5)$$

For,**Day-4**

$$\text{Prior} = \lambda \sim \text{Gamma}(108,5)$$
$$\text{Posterior} = \lambda \sim \text{Gamma}(108+27,5+1) = \text{Gamma}(135,6)$$

So,Prior Distribution for Day-5 is same as Posterior Distribution of Day-4

$$\mathbf{Prior = \lambda \sim Gamma(135,6)}$$

3.2)
Road accidents are predicted to happen on day 5,

We can calculate the expected value of λ from the Gamma distribution, which represents the average rate of accidents,which is given as

$$135/6 = 22.5$$

Thus, the predicted number of road accidents on Day 5 is approximately 22.5.

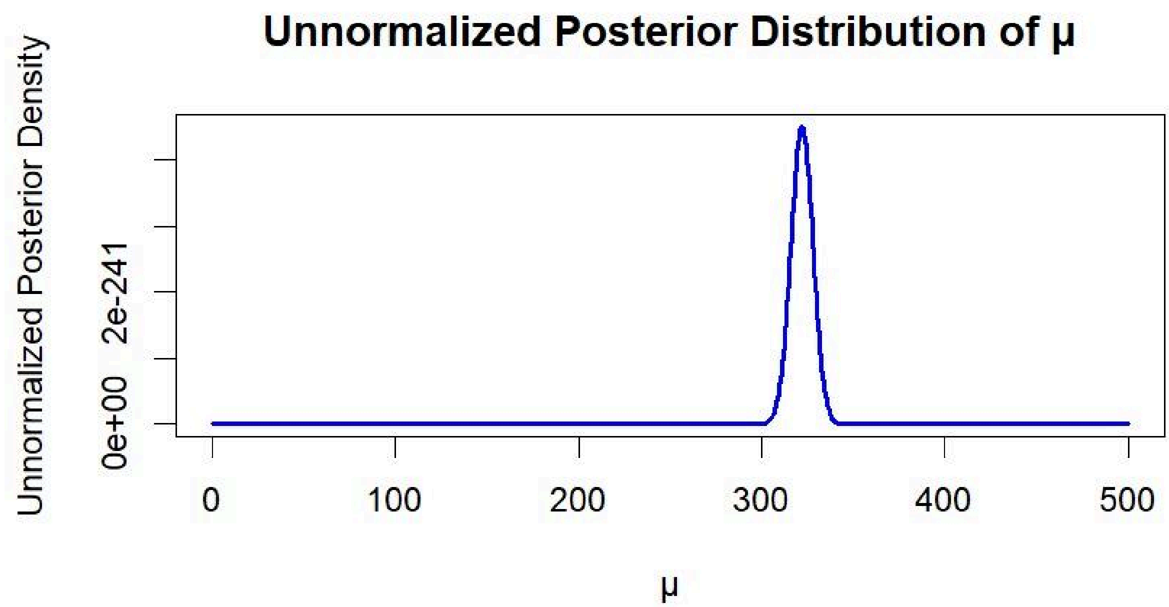
Part-4:Model building in the Bayesian framework

4.5.1)

R code used to plot the graph :

```
# Reading the data from github
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

# sigma value
sigma <- 60

# Define µ values
µ <- seq(0, 500, length.out = 1000)

# Function to compute unnormalized posterior density for given µ
and δ
compute_unnormalized_posterior <- function(µ, δ) {
  term1 <- prod(dnorm(tw, mean = µ, sd = sigma))
  term2 <- prod(dnorm(tnw, mean = µ + δ, sd = sigma))
  term3 <- dnorm(µ, mean = 300, sd = 50)
  return(term1 * term2 * term3)
}

# Compute unnormalized posterior density for each µ in µ
unnormalized_posterior <- sapply(µ, function(m) {
  compute_unnormalized_posterior(m, 0)  # Assuming δ = 0 for
simplicity
})

# Plot the unnormalized posterior distribution
plot(µ, unnormalized_posterior, type = "l",
     xlab = "µ", ylab = "Unnormalized Posterior Density",
```

```
main = "Unnormalized Posterior Distribution of μ",
col = "blue", lwd = 2)
```

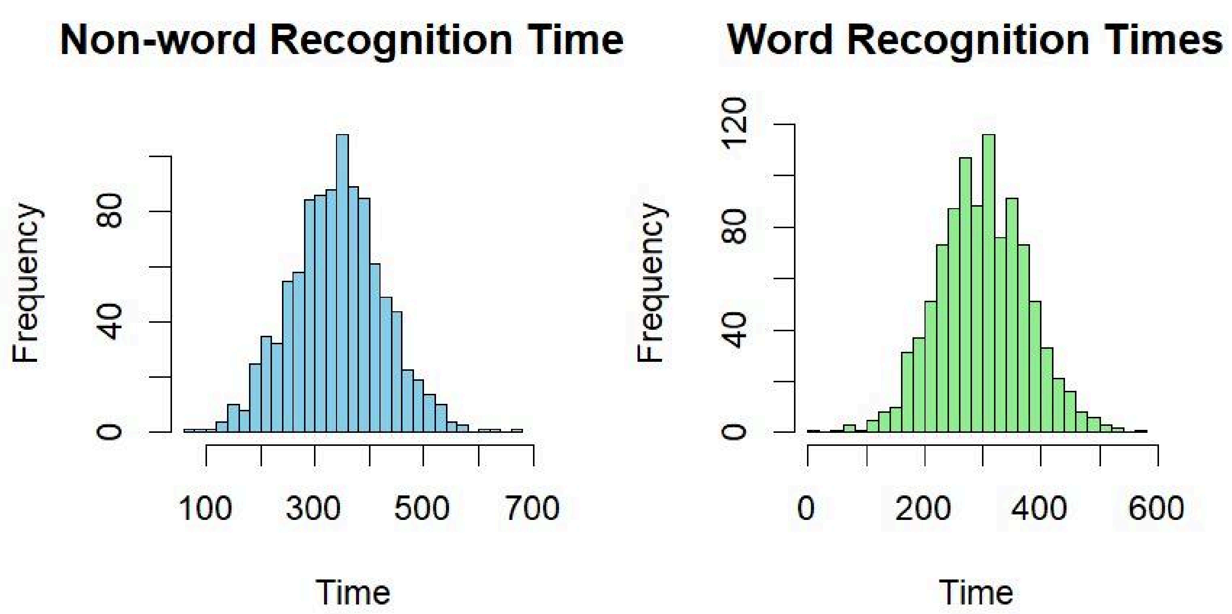


4.5.2)

R Code used to solve the problem:

```
# Load necessary libraries
library(truncnorm)

#sigma value
sigma <- 60

# Step 1: Generate μ from N(300, 50)
set.seed(123)
num_samples <- 1000
mu_prior <- rnorm(num_samples, mean = 300, sd = 50)

# Step 2: Generate δ from N+(0, 50)
delta_prior <- rtruncnorm(num_samples, a = 0, b = Inf, mean = 0,
sd = 50)
```

```
# Step 3: Generate recognition times
# Non-word recognition times: N(μ + δ, σ = 60)
nonword_recognition_times <- rnorm(num_samples, mean = mu_prior
+ delta_prior, sd = sigma)

# Word recognition times: N(μ, σ = 60)
word_recognition_times <- rnorm(num_samples, mean = mu_prior, sd
= sigma)

# Step 4: Plot recognition times as histograms
par(mfrow = c(1, 2))

# Histogram for non-word recognition times
hist(nonword_recognition_times, breaks = 30, col = "skyblue",
     main = "Non-word Recognition Times", xlab = "Time")

# Histogram for word recognition times
hist(word_recognition_times, breaks = 30, col = "lightgreen",
     main = "Word Recognition Times", xlab = "Time")
```



4.5.3)

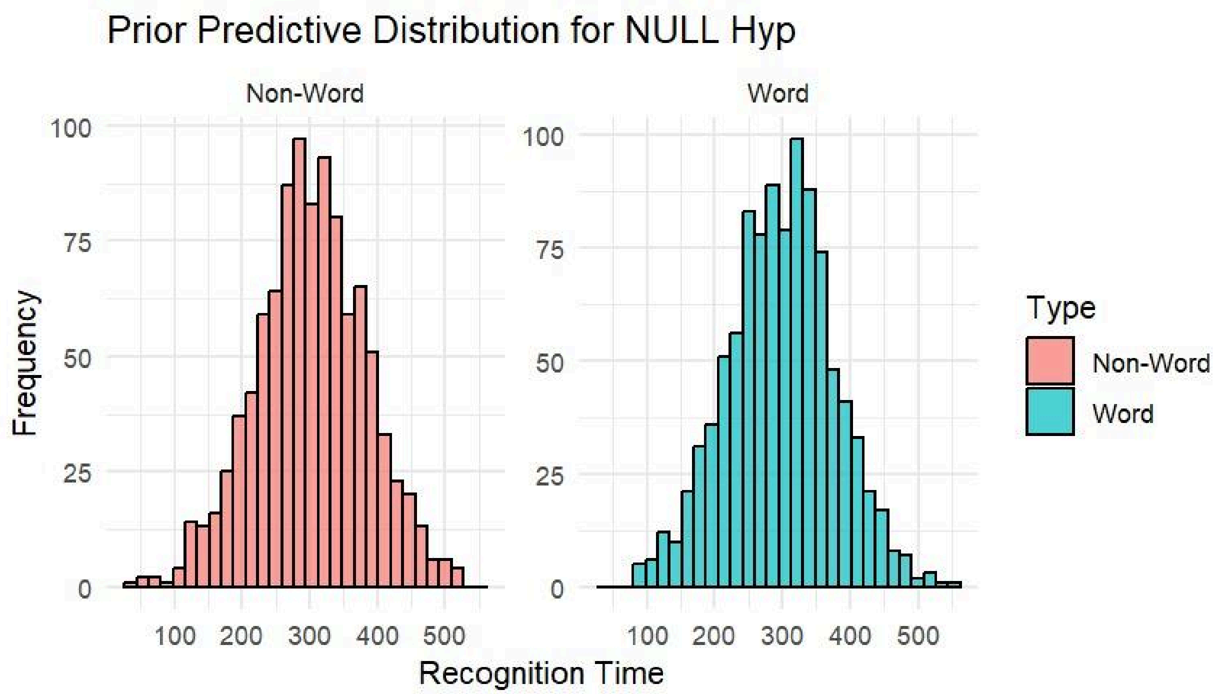
R code to generate prior prediction for null hypothesis :

```
# Load necessary library
library(ggplot2)

# Parameters
mu_prior_mean <- 300
mu_prior_sd <- 50
sigma <- 60
delta <- 0
num_samples <- 1000

# Generate samples from the prior distribution of mu
mu_samples <- rnorm(num_samples, mu_prior_mean, mu_prior_sd)

# Generate word recognition times (Tw) and non-word recognition
times (Tnw)
Tw_samples <- rnorm(num_samples, mu_samples, sigma)
Tnw_samples <- rnorm(num_samples, mu_samples + delta, sigma)

# Create data frames for plotting
Tw_df <- data.frame(Time = Tw_samples, Type = 'Word')
Tnw_df <- data.frame(Time = Tnw_samples, Type = 'Non-Word')
data <- rbind(Tw_df, Tnw_df)

# Plot the histograms
ggplot(data, aes(x = Time, fill = Type)) +
  geom_histogram(alpha = 0.7, position = 'identity', bins = 30,
color = 'black') +
  facet_wrap(~ Type, scales = 'free_y') +
  labs(title = 'Prior Predictive Distribution for NULL Hyp',
       x = 'Recognition Time',
       y = 'Frequency') +
  theme_minimal()
```

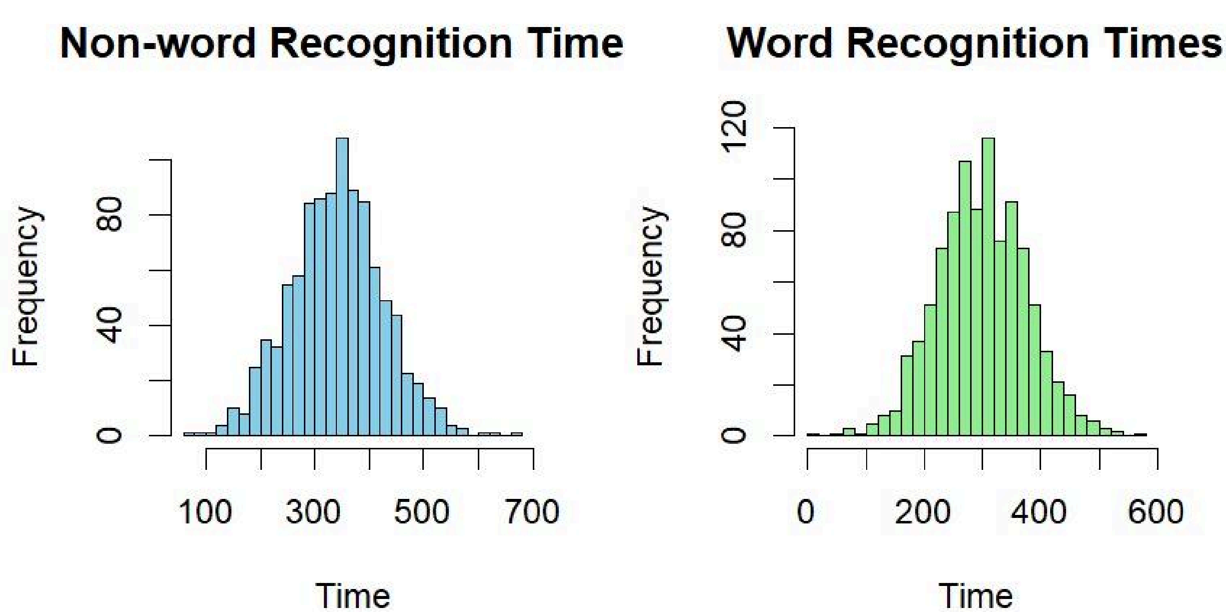
Prior predictions from the lexical-access model already calculated in the previous part.

Comparison between priors of both the models:

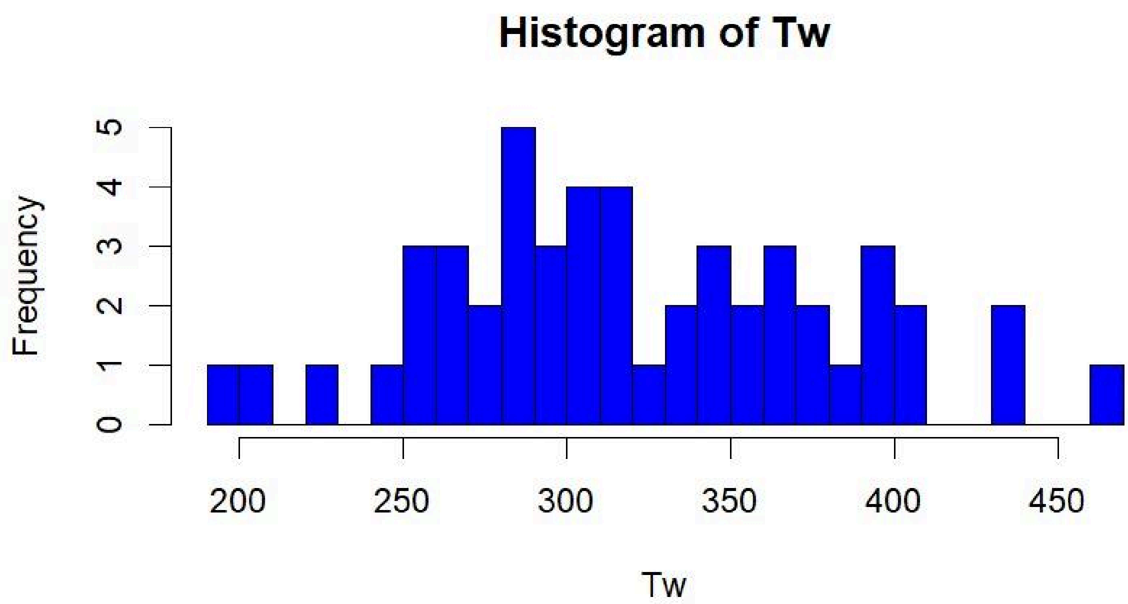
NULL Hypothesis:

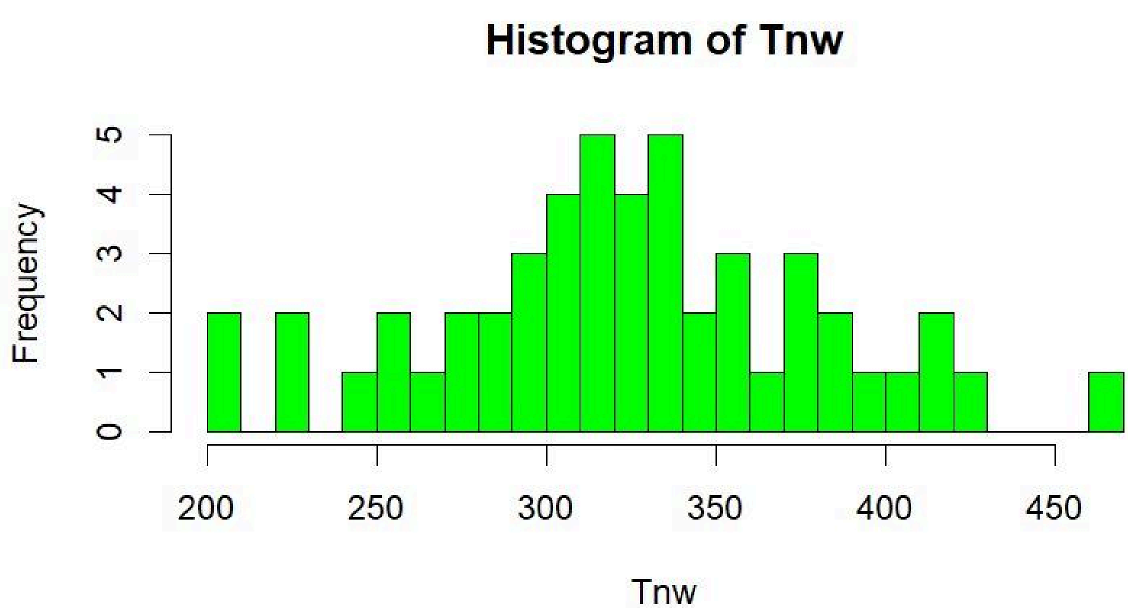


Lexical-access model:



4.5.4 solution: (Compare the prior predictions of each model against the observed data Tw and Tnw. Which model seems more consistent with the data ?)
Observed data:





On comparison of the original data from that of null hypothesis and lexical-access model we observed that lexical-access model fits the observed data better.

4.5.5)

R Code for plotting:

```
# Load necessary libraries
library(truncnorm)


# Read data
dat <- read.table(

"https://raw.githubusercontent.com/yadavhimanshu059/CGS698C/main
/notes/Module-2/recognition.csv",
  sep = ",", header = TRUE)[, -1]

tw <- dat$Tw
tnw <- dat$Tnw


# Assuming sigma value
sigma <- 60


# Define μ and δ values
μ <- seq(0, 500, length.out = 1000)
δ_vals <- seq(0, 200, length.out = 100)


# Function to compute unnormalized posterior density for given μ
and δ
compute_unnormalized_posterior <- function(μ, δ) {
  term1 <- prod(dnorm(tw, mean = μ, sd = sigma))
  term2 <- prod(dnorm(tnw, mean = μ + δ, sd = sigma))
  term3 <- dnorm(μ, mean = 300, sd = 50)
  term4 <- dtruncnorm(δ, a = 0, b = Inf, mean = 0, sd = 50)
  return(term1 * term2 * term3 * term4)
}


# Compute unnormalized posterior density for each δ in δ_vals
```

```
unnormalized_posterior_δ <- sapply(δ_vals, function(δ) {
  sum(sapply(μ, function(m) {
    compute_unnormalized_posterior(m, δ)
  }))
})


# Plot the unnormalized posterior distribution
plot(δ_vals, unnormalized_posterior_δ, type = "l",
     xlab = expression(delta), ylab = "Unnormalized Posterior
Density",
     main = "Unnormalized Posterior Distribution of δ",
     col = "blue", lwd = 2)
```