Assignment 2

Aryan Thada : 210205

2024-06-12

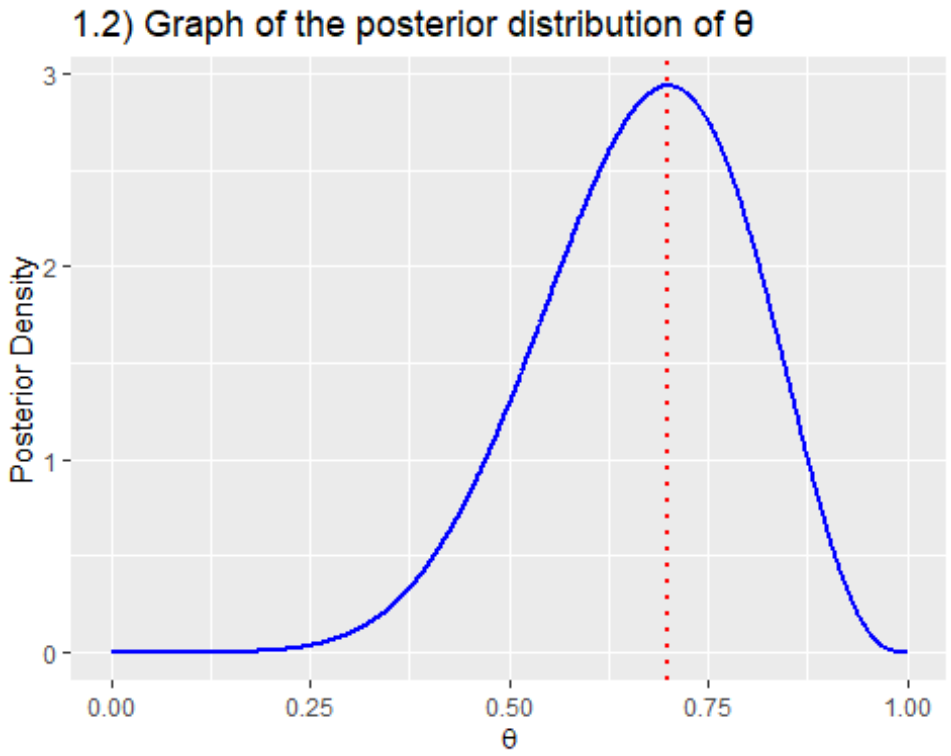
{Q1}

```
library(reshape2)
library(ggplot2)

lkd <- function(theta , y)
{
  x<- factorial(10)/(factorial(y)  * factorial(10-y))
  lkd_value <- x * (theta^y)*((1-theta)^(10-y))
  return (lkd_value)
}

prior <- function(x) {
  ans <- ifelse(x >= 0 & x <= 1, 1, 0)
  return(ans)
}

posterior <- function(x, y) {
  posterior_density <- (lkd(x, y) * prior(x))/(1/11);
  return(posterior_density)
}

#cat("1.1 a)",posterior(0.75, 7),"\n")
#cat("1.1 b)",posterior(0.25, 7),"\n")
#cat("1.1 c)",posterior(1, 7),"\n")

y<- 7
n<- 10
marginal_lkd <-  1/11

theta_values <- c(0.75,0.25,1)

posterior_densities <- sapply(theta_values, posterior,y=y)
result <- data.frame(theta = theta_values, posterior_density =
posterior_densities)
print(result)
```

```
##   theta posterior_density
## 1  0.75        2.75310516
## 2  0.25        0.03398895
## 3  1.00        0.00000000
```

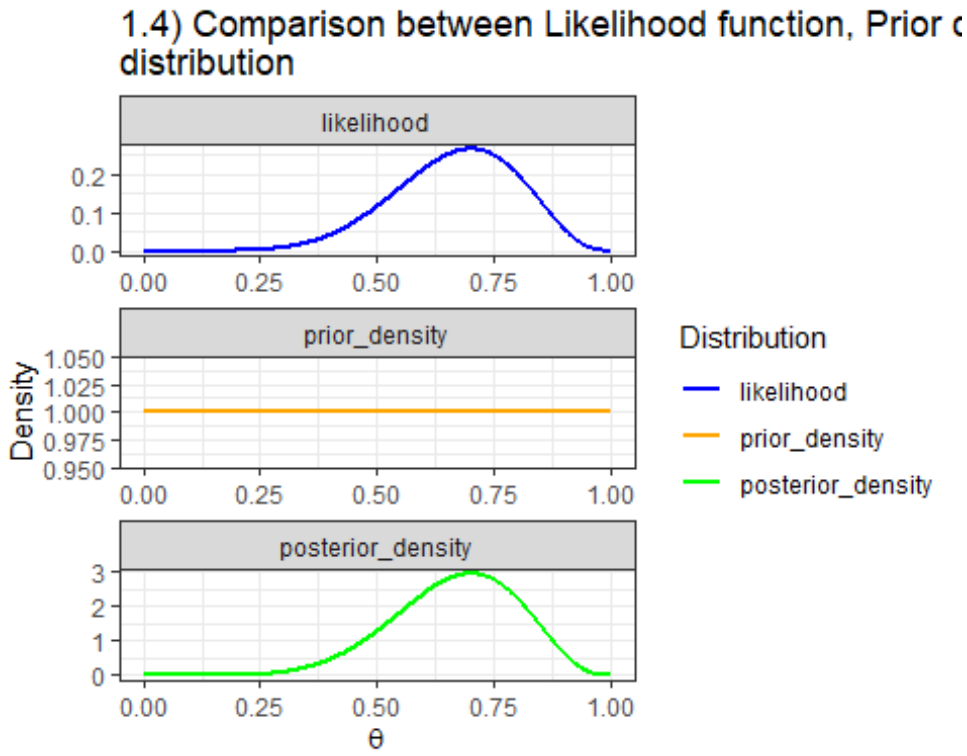
```r
theta_seq <- seq(0,1,length.out = 1000)

posterior_density_seq <- sapply(theta_seq,posterior,y=y)

df <- data.frame(theta = theta_seq , posterior_density =
posterior_density_seq)

ggplot(df,aes(x = theta ,y = posterior_density))+
  geom_line(size = 1,color = "blue") +
  geom_vline(xintercept = theta_seq[which.max(posterior_density_seq)],
linetype = "dotted", color = "red",
             linewidth=1)+
  xlab(expression(theta)) +
  ylab("Posterior Density") +
  labs(title = "1.2) Graph of the posterior distribution of θ")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```r
max_posterior_theta <- theta_seq[which.max((posterior_density_seq))]
cat("The value of theta with the maximum posterior density is:",
max_posterior_theta, "and maximum posterior density
=",max(posterior_density_seq),"\n")
```

```
## The value of theta with the maximum posterior density is: 0.6996997 and
maximum posterior density = 2.935101

df <- data.frame(theta = theta_seq, likelihood = lkd(theta_seq, 7),
prior_density = prior(theta_seq),
                 posterior_density = posterior(theta_seq, 7))
df.m <- melt(df, id.vars = "theta", variable.name = "Distribution")


ggplot(df.m, aes(x = theta, y = value, color = Distribution)) +
  geom_line(size = 1) +
  facet_wrap(~Distribution, scales = "free", nrow = 3) +
  scale_color_manual(values = c("likelihood" = "blue",
                                "prior_density" = "orange",
                                "posterior_density" = "green")) +
  theme_bw() +
  xlab(expression(theta)) +
  ylab("Density") +
  labs(title = "1.4) Comparison between Likelihood function, Prior
distribution and Posterior
distribution")
```




```
#theta posterior_density
#1  0.75        2.75310516
#2  0.25        0.03398895
#3  1.00        0.00000000
```

```
#The value of theta with the maximum posterior density is: 0.6996997 and
maximum posterior density = 2.935101
```

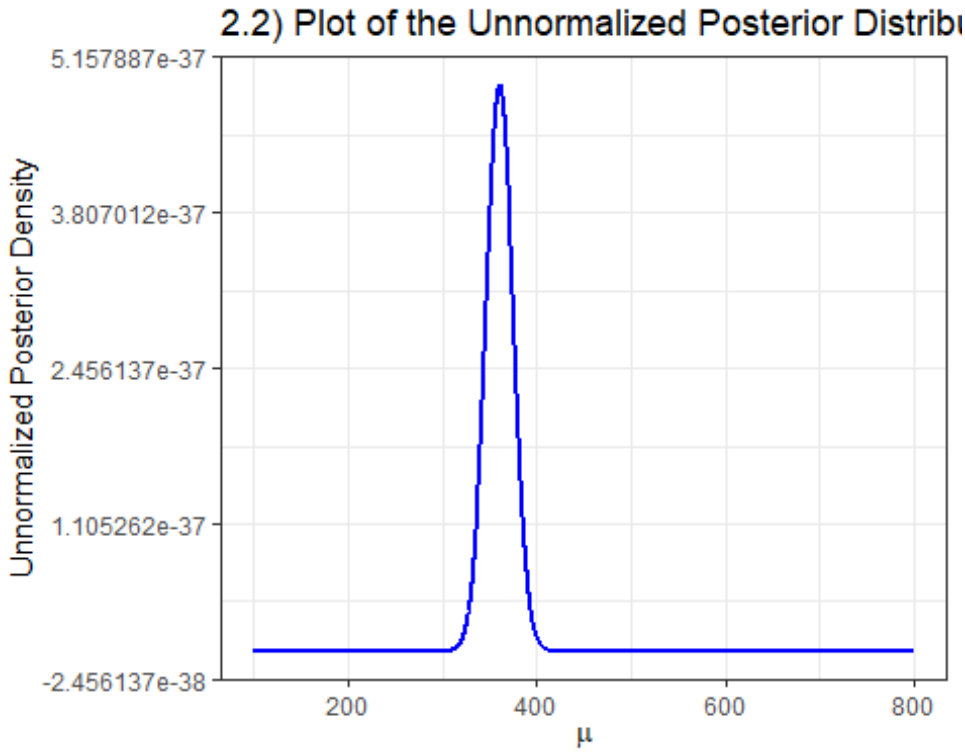
{Q2}

```
library(reshape2)
library(ggplot2)
y <- c(300, 270, 390, 450, 500, 290, 680, 450)
n <- length(y)

sigma <- 50

mu_prior_mean <- 250
mu_prior_sd <- 25

lkd_func <- function(mu, x, y) {
  p <- 1 / ((x * sqrt(2 * pi)) ^ (length(y)));
  lkd_val <- p * exp((-1 / (2 * x^2)) * sum((y - mu)^2));
  return(lkd_val)
}

prior2 <- function(mu) {
  dnorm(mu, mean = mu_prior_mean, sd = mu_prior_sd);
}



unn_post <- function(mu, x, y) {
  ans <- lkd_func(mu, x, y) * prior2(mu);
  return(ans);
}

mu_values <- c(300, 900, 50)
posterior_densities <- sapply(mu_values, unn_post, x = sigma, y = y)
result <- data.frame(mu = mu_values, unnormalized_posterior_density =
posterior_densities)
print(result)
```

```
##    mu unnormalized_posterior_density
## 1 300                   6.824248e-41
## 2 900                   0.000000e+00
## 3  50                  9.691374e-138
```

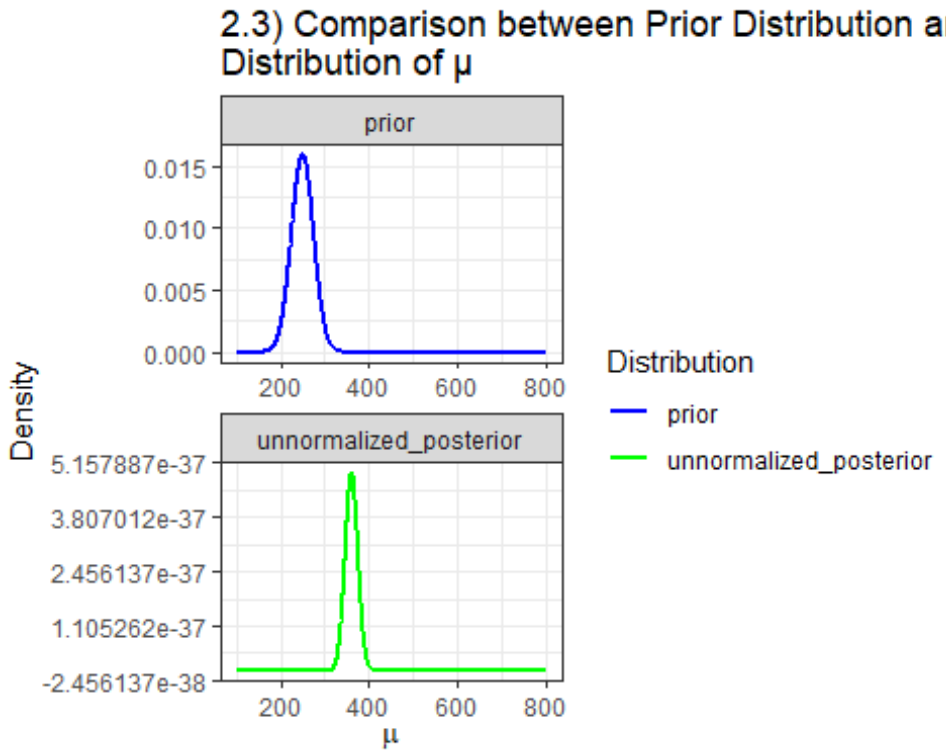
```
#cat("2.1 a) For μ = 300 :",unn_post(300, 50, y), "\n");
#cat("2.1 b) For μ = 900 :",unn_post(900, 50, y), "\n");
#cat("2.1 c) For μ = 50 :",unn_post(50, 50, y), "\n");

mu_seq <- seq(100, 800, length.out = 1000)

posterior_density <- sapply(mu_seq, unn_post, x = sigma, y = y)
```

```r
df <- data.frame(mu = mu_seq, posterior_density = posterior_density )


ggplot(df, aes(x = mu, y = posterior_density )) +
  geom_line(size = 1, color = "blue") +
  theme_bw() +
  xlab(expression(mu)) +
  ylab("Unnormalized Posterior Density") +
  labs(title = "2.2) Plot of the Unnormalized Posterior Distribution of μ")
```




```r
prior_density <- dnorm(mu_seq, mean = 250, sd = 25);
df <- data.frame(mu = mu_seq,
                 prior = prior_density,
                 unnormalized_posterior = posterior_density)
df.m <- melt(df, id.vars = "mu", variable.name = "Distribution")
ggplot(df.m, aes(x = mu, y = value, color = Distribution)) +
  geom_line(size = 1) +
  facet_wrap(~Distribution, scales = "free", nrow = 2) +
  scale_color_manual(values = c("prior" = "blue",
                                "unnormalized_posterior" = "green")) +
  theme_bw() +
  xlab(expression(mu)) +
  ylab("Density") +
  labs(title = "2.3) Comparison between Prior Distribution and Unnormalized
```

```
Posterior
Distribution of μ")
```



{Q3}

Likelihood Assumptions->

The number of road accidents on ith day :

ki ~poisson(lambda)

Prior assumption for day 1 : lambda ~ Gamma(40,2)

Posterior after day 1: (lambda|k1) ~ Gamma(40+k1,3)

Using Bayesian method Posterior of day 1 will now become prior for day 2.

Prior distribution for lambda after day 1 and before day 2 : (lambda)= (lambda|k1) ~ Gamma(40+k1,3)

Posterior distribution of lambda after day 2 : (lambda|k2) ~ Gamma(40+k1+k2,4)

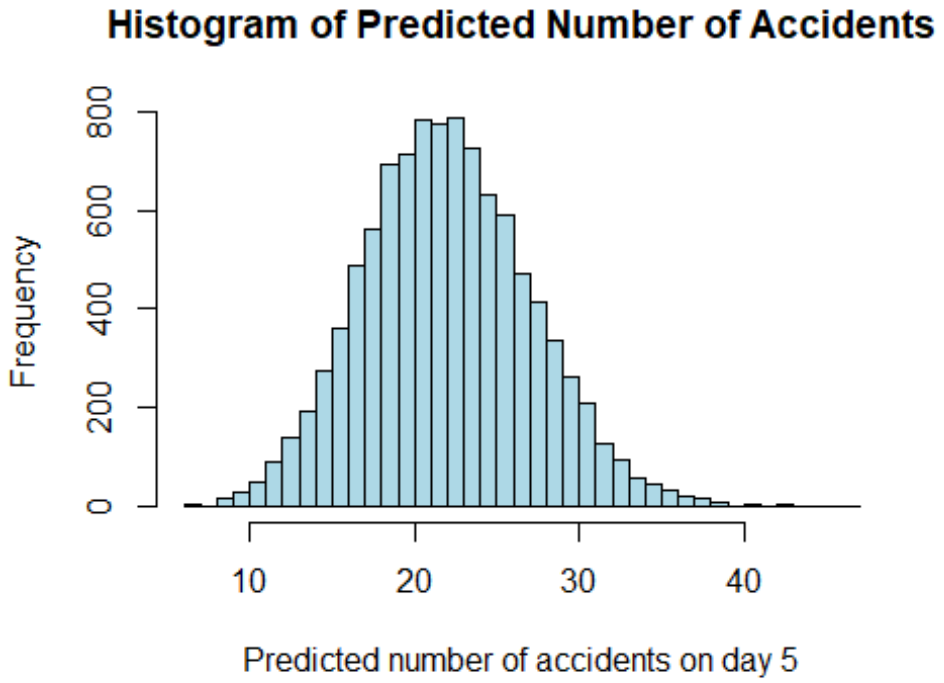
Similarly we can find priors to ith day which will be equal to to posterior of (i-1)th day and new posterior after ith day.

(3.1) The prior on lambda to generate predictions for day 5 :

(lambda) ~Gamma(40+25+20+23+27,2+1+1+1+1)

(lambda) ~ Gamma(135,6)

```
lambda <- rgamma(10000,shape = 135,rate = 6)

k_pred <- numeric(10000)
for(i in 1:10000)
{
  k_pred[i] <- rpois(1,lambda[i])
}

hist(k_pred, xlab = "Predicted number of accidents on day 5", main =
"Histogram of Predicted Number of Accidents", breaks = 50, col = "lightblue")
```



{Q4}

Null hypothesis model :-

Tw is the vector of word recognition times, and Tnw is the vector of non-word recognition times.

Likelihood assumptions ->

Tw ~ Normal(mu,sigma)

Tnw ~ Normal(mu + delta , sigma)

Priors assumptions->

mu ~ Normal(300,50)

sigma = 60

delta = 0

Lexical - access model :-

Tw is the vector of word recognition times, and Tnw is the vector of non-word recognition times.

Likelihood assumptions ->

Tw ~ Normal(mu,sigma)

Tnw ~ Normal(mu + delta , sigma)

Priors assumptions->

mu ~ Normal(300,50)

sigma = 60

delta ~Normal_+(0,50)

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

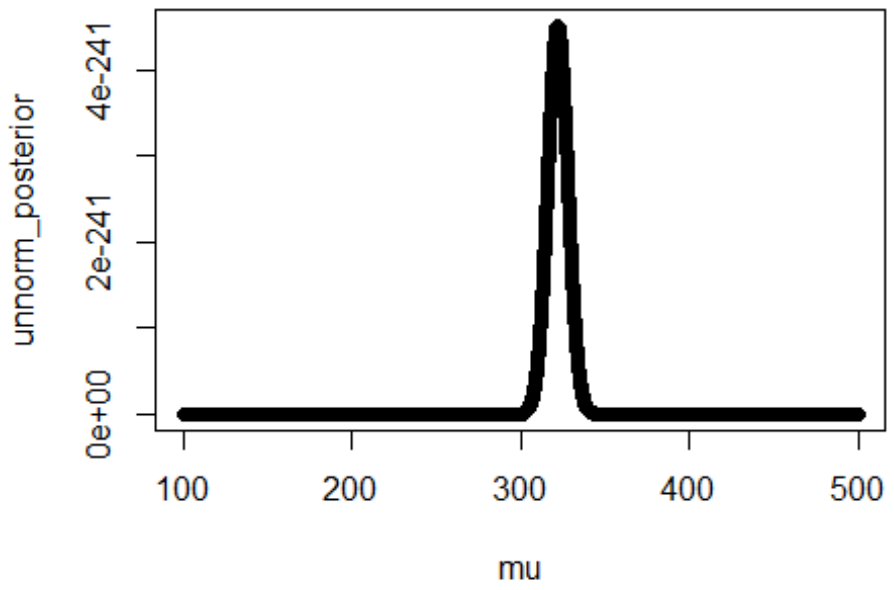
```
##         Tw      Tnw
## 1 285.0780 296.8060
## 2 267.5184 280.1157
## 3 289.9203 310.4417
## 4 399.0674 324.8276
## 5 359.9884 373.8152
## 6 403.3993 269.8220
```

```
## Tw Tnw
## 1 285.0780 296.8060
## 2 267.5184 280.1157
## 3 289.9203 310.4417
## 4 399.0674 324.8276
## 5 359.9884 373.8152
## 6 403.3993 269.8220
```

[4.5.1]

```
mu <- runif(50000,100,500)
sigma <- 60
delta <- 0
```

```
lkhood <- rep(NA,50000)
prior <- rep(NA,50000)
unnorm_posterior <- rep(NA,50000)
for(i in 1:50000){
lkhood[i] <- prod(dnorm(dat$Tw,mu[i],sigma)*
dnorm(dat$Tnw,mu[i]+delta,sigma))
prior[i] <- dnorm(mu[i],300,50)
unnorm_posterior[i] <- lkhood[i]*prior[i]
}
plot(mu,unnorm_posterior)
```

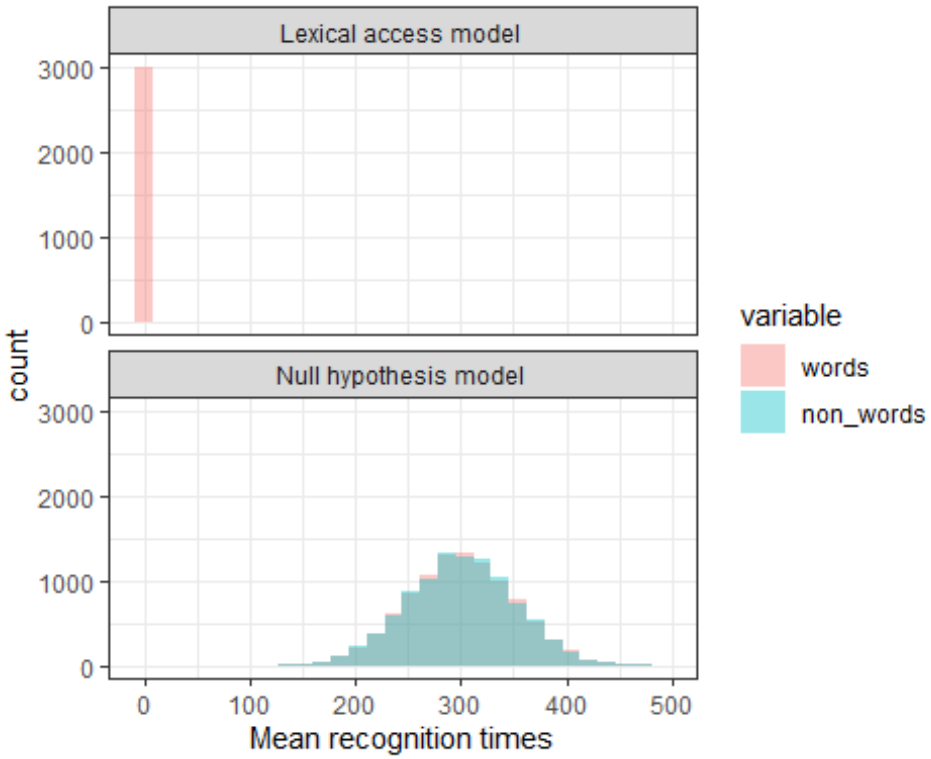


[4.5.2]

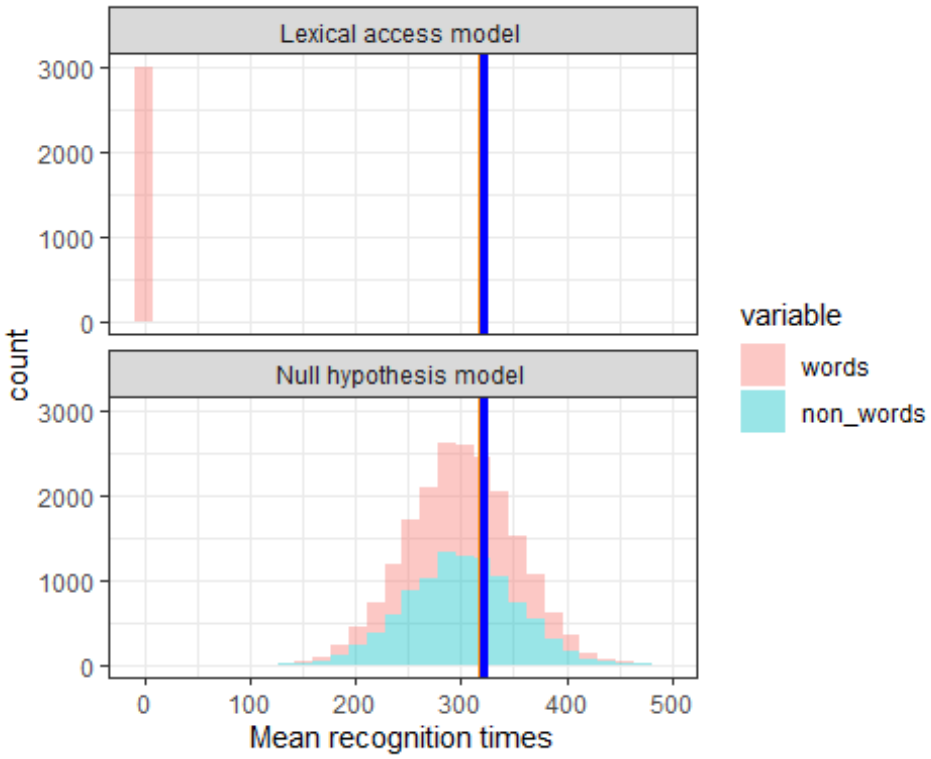
```
library(data.table)

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:reshape2':
##
##     dcast, melt

library(ggplot2)
library(reshape2)
library(dplyr)

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

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

mu <- rnorm(10000, 300, 50)
sigma <- rep(60, 10000)
delta <- rep(0, 10000)

n<- length(dat$Tw)

# Preallocate list to store dataframes
list_df <- vector("list", 10000)

# Generate data
for (i in 1:10000) {
  Tw_pred <- rnorm(n, mu[i], sigma[i])
  Tnw_pred <- rnorm(n, mu[i] + delta[i], sigma[i])

  list_df[[i]] <- data.table(
    sample_id = rep(i, n),
    mu = rep(mu[i], n),
    delta = rep(delta[i], n),
    obs_id = 1:n,
    Tw_pred = Tw_pred,
    Tnw_pred = Tnw_pred
  )
}


df.pred <- rbindlist(list_df)

df.pred.mean <- df.pred[, .(words = mean(Tw_pred), non_words =
mean(Tnw_pred)), by = sample_id]
df.pred.mean[, model := "Null hypothesis model"]

df.pred.null.hypothesis <- melt(df.pred.mean, id.vars = c("model",
"sample_id"))


df.pred.lexical.access <- data.table(model = "Lexical access model",
sample_id = 1:3000, variable = "words", value = rnorm(3000))
```

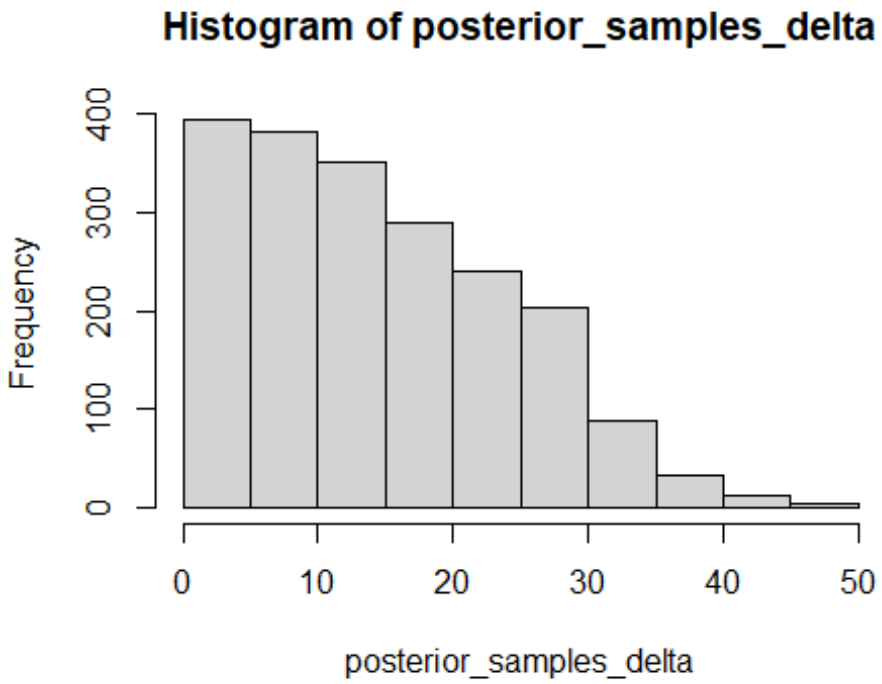
```
df.pred <- rbind(df.pred.lexical.access, df.pred.null.hypothesis, fill =
TRUE)

ggplot(df.pred, aes(x = value, group = variable, fill = variable)) +
  geom_histogram(alpha = 0.4, position = "identity") +
  xlab("Mean recognition times") + theme_bw() +
  facet_wrap(~model, nrow = 2)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(df.pred, aes(x=value,group=variable,fill=variable))+
geom_histogram(alpha=0.4)+
xlab("Mean recognition times")+theme_bw()+
facet_wrap(~model,nrow=2)+
geom_vline(xintercept=mean(dat$Tw),color="darkorange",size=1.5)+
geom_vline(xintercept = mean(dat$Tnw),color="blue",size=1.5)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
library(truncnorm)
mu <- runif(50000,100,500)
sigma <- 60
delta <- runif(50000,0,50)

likelihood <- rep(NA,50000)

prior <- rep(NA,50000)

posterior_unnorm <- rep(NA,50000)

for(i in 1:50000)
{
    likelihood[i] <- prod(dnorm(dat$Tw,mu[i],sigma))*
    prod(dnorm(dat$Tnw,mu[i]+delta[i],sigma))
    prior[i] <- dnorm(mu[i],300,50)*
    dtruncnorm(delta[i],a=0,b=Inf,mean=0,sd=50)
    posterior_unnorm[i] <- likelihood[i]*prior[i]
}
posterior_samples_delta <- sample(delta,size=2000,prob = posterior_unnorm)


hist(posterior_samples_delta)
```


Histogram of posterior_samples_delta
Frequency
0
100
200
300
400
0
10
20
30
40
50
posterior_samples_delta