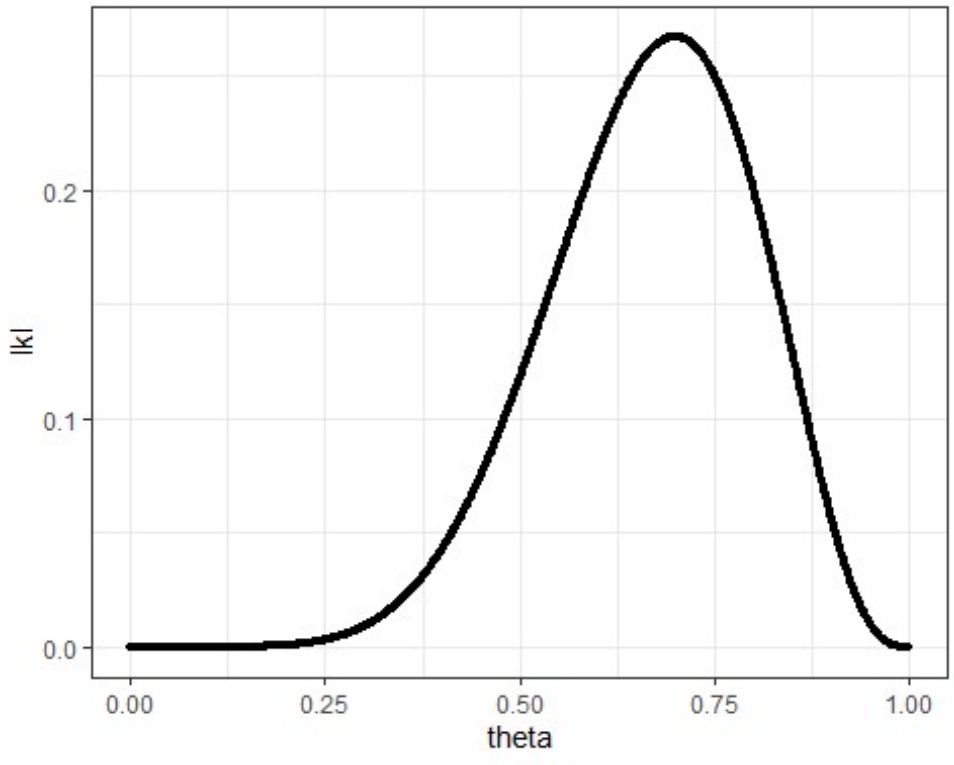
Assignment 2 Amipriya Anand 220122.R

Amipriya Ananad

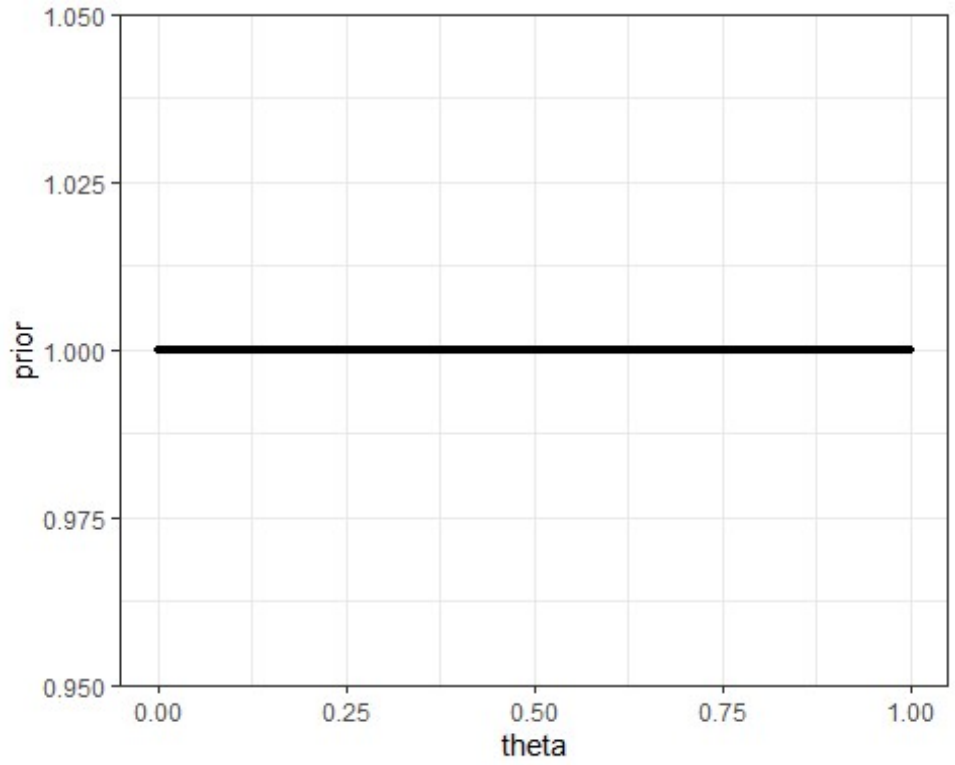
(220122)

2024-06-14

```
### **Assignment 2**

## **Part 1:  A simple binomial model**

x = 7
n = 10

theta = seq(from = 0,to =1, length = 1000)

likelihoods = dbinom(x = 7, size = 10, prob = theta)

data = data.frame(theta = theta,lkl = likelihoods)

library(ggplot2)

ggplot(data,aes(theta,lkl))+
  geom_point(size = 1)+
  theme_bw()
```



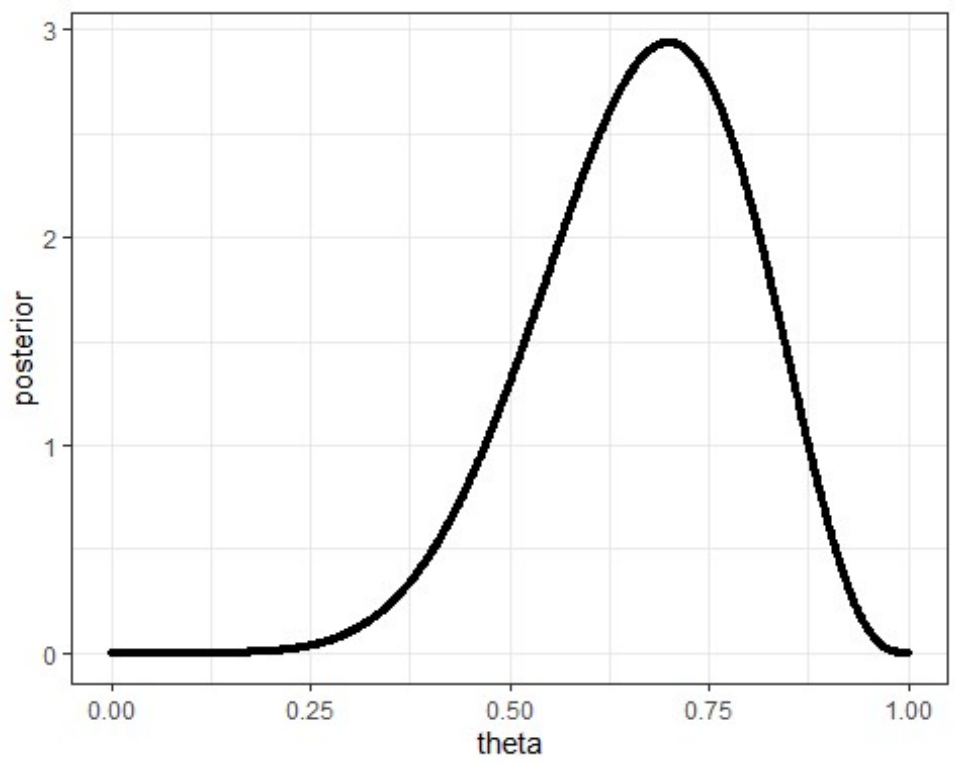

```r
#priors

data$prior = ifelse(theta<=1,1,0)
ggplot(data,aes(theta,prior))+
  geom_point(size =1)+
  theme_bw()
```




```
#marginal likelihood (given)
ml = 1/11

#posterior
data$posterior = (data$lkl*data$prior)/ml


#Ans 1.1
theta_given = c(0.25,0.75,1)
for(i in theta_given)
{
  print(dbinom(7,10,i)*1*11)
}
```

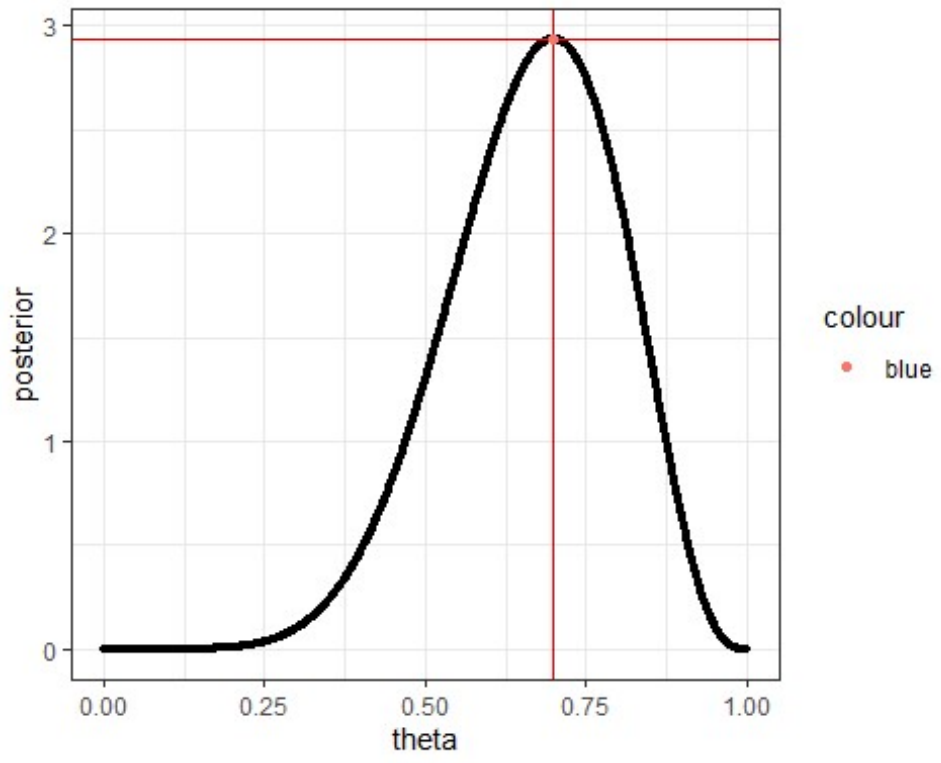
```
## [1] 0.03398895
## [1] 2.753105
## [1] 0
```

```
#Ans 1.2
ggplot(data,aes(theta,posterior))+
  geom_point(size =1)+
  theme_bw()
```

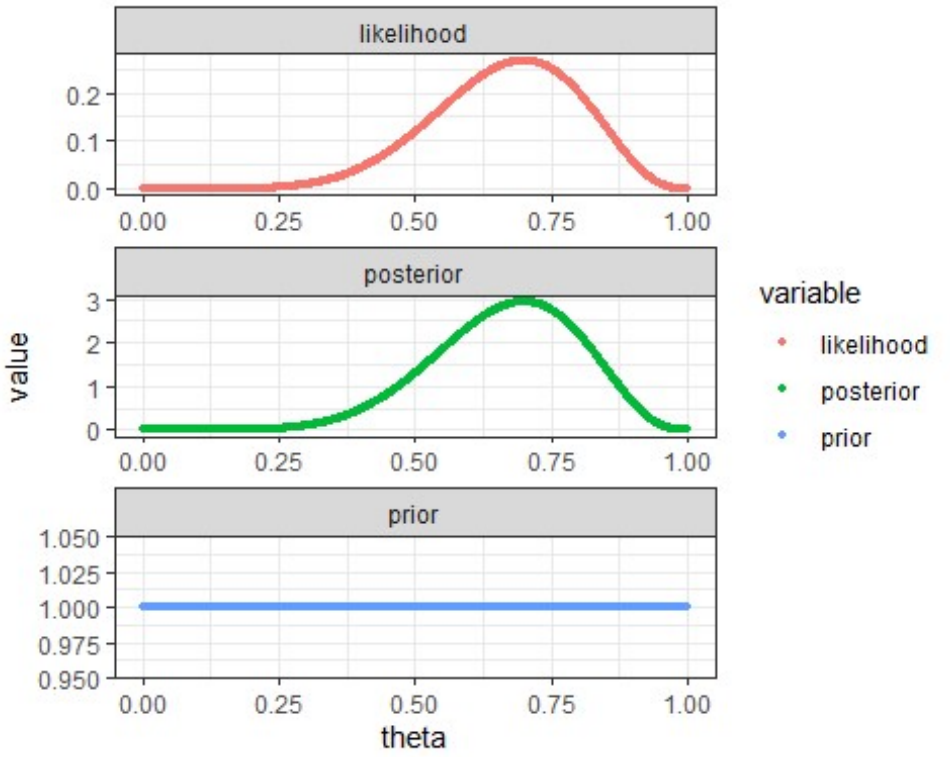



```r
#Ans 1.3
post_max = max(data$posterior)
theta_max = data$theta[which(data$posterior == post_max)]

ggplot(data,aes(theta,posterior))+
  geom_point(size =1)+
  theme_bw()+
  geom_vline(xintercept = theta_max, color = "red")+
  geom_hline(yintercept = post_max, color = "red")+
  geom_point(aes(theta_max,post_max, color = "blue"))
```

```
## Warning in geom_point(aes(theta_max, post_max, color = "blue")): All
aesthetics have length 1, but the data has 1000 rows.
## ℹ Please consider using `annotate()` or provide this layer with data
containing
##   a single row.
```




```r
# Ans 1.4

library(reshape2)

data.m = melt(data, id = ("theta"))
data.m$variable = ifelse(data.m$variable=="lkl","likelihood",

ifelse(data.m$variable=="posterior","posterior","prior"))


ggplot(data.m, aes(theta,value,group = variable, colour = variable))+
  geom_point(size =1)+
  theme_bw()+
  facet_wrap(~variable,scales="free",nrow=3)
```

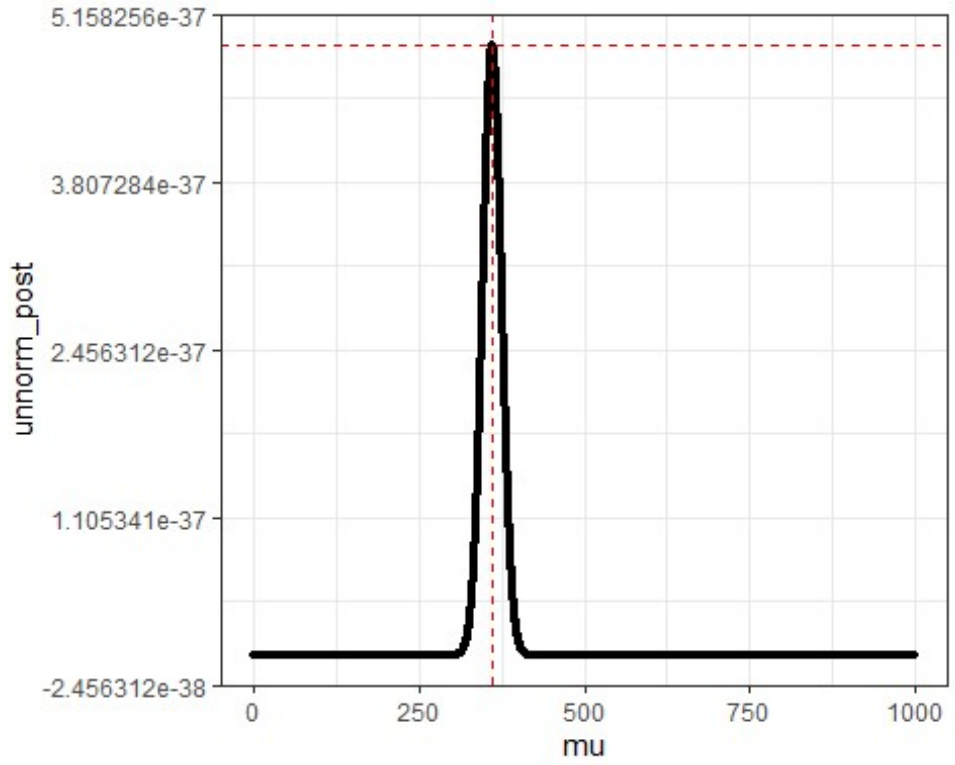

likelihood
posterior
prior
value
theta
variable
likelihood
posterior
prior
0.0
0.1
0.2
0
1
2
3
0.950
0.975
1.000
1.025
1.050
0.00
0.25
0.50
0.75
1.00

```
## **Part 2: A Gaussian model of reading**

y = c(300, 270, 390, 450, 500, 290,
      680, 450)
sigma = 50
mu =  seq(from =0, to = 1000, length = 10000)
data2 = data.frame(mu = mu, sigma = sigma)
data2$lkl = rep(NA, length(mu))
for(i in 1 :length(mu))
{
  data2$lkl[i] = prod(dnorm(x = y,mean = mu[i],sd = sigma))
}

# priors
prior = dnorm(mu, 250,25)
data2$prior = prior
data2$unnorm_post = data2$lkl*data2$prior

#Ans 2.1
mu_given = c(300,900,50)
for(i in mu_given)
{
  print((prod(dnorm(y,i,sigma)))*dnorm(i,250,25))
}
```

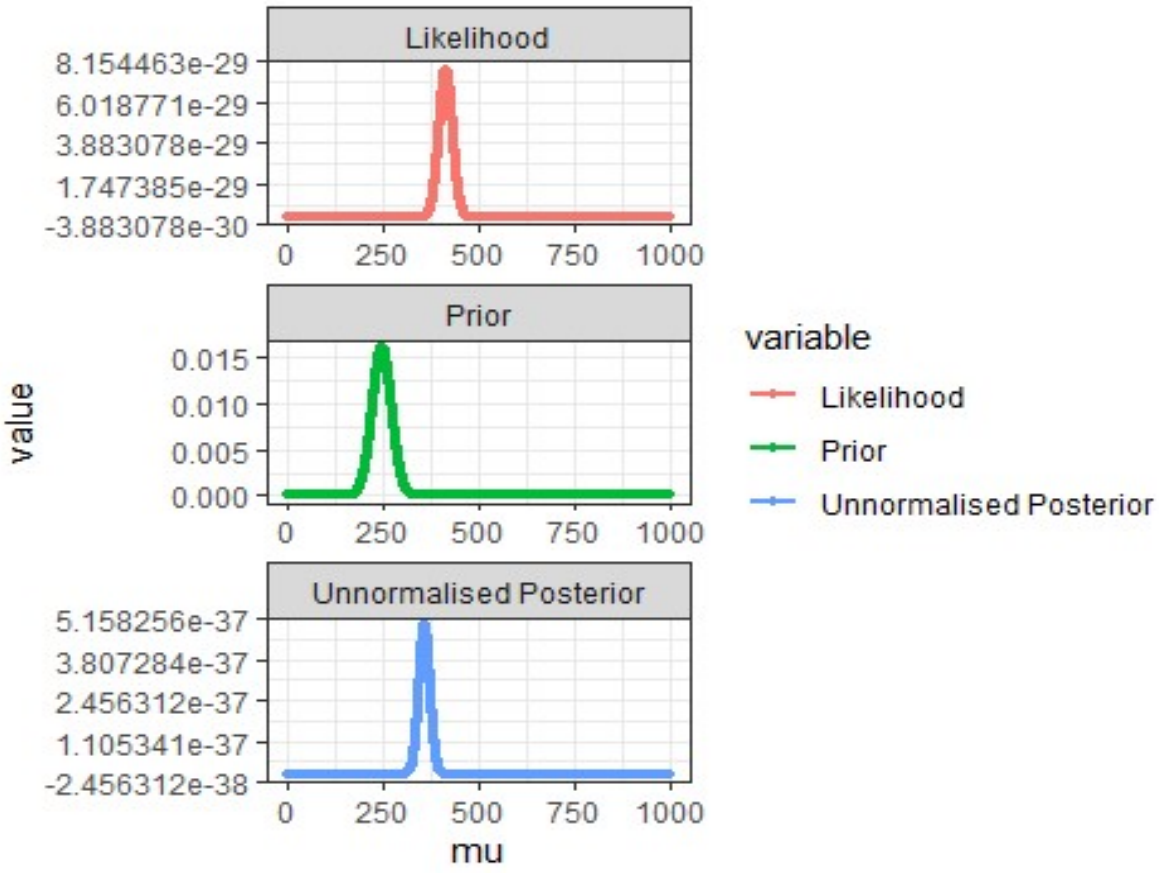
```
## [1] 6.824248e-41
## [1] 0
## [1] 9.691374e-138
```

```
#Ans 2.2
ggplot(data2, aes(mu,unnorm_post))+
  geom_point(size =1)+
  geom_line(size =1)+
  theme_bw()+
  geom_hline(yintercept = max(data2$unnorm_post),size = 0.5, linetype =
"dashed", color = "red")+
  geom_vline(xintercept = data2$mu[which(data2$unnorm_post ==
max(data2$unnorm_post))],
             size = 0.5,linetype = "dashed",color = "red")
```

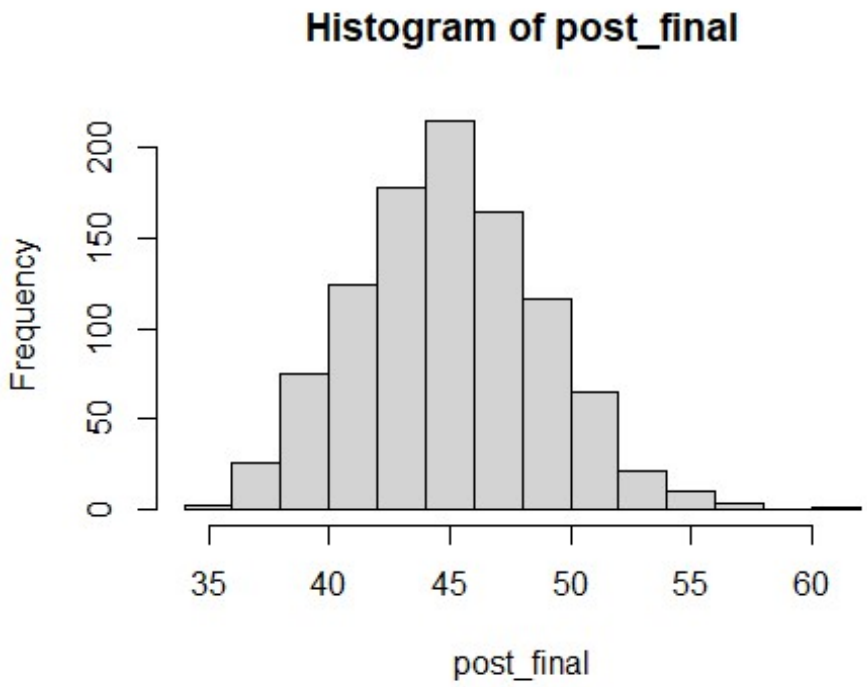
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
#Ans 2.3
data2.m = melt(data2,id = c("mu", "sigma"))
data2.m$variable = ifelse(data2.m$variable=="lkl","Likelihood",

ifelse(data2.m$variable=="prior","Prior","Unnormalised Posterior"))
ggplot(data2.m,aes(mu,value,goup = variable,colour = variable))+
  geom_point(size = 1)+
  geom_line(size =1 )+
  theme_bw()+
  facet_wrap(~variable, scales = "free",nrow = 3)
```


Likelihood
8.154463e-29
6.018771e-29
3.883078e-29
1.747385e-29
-3.883078e-30
0
250
500
750
1000
Prior
0.015
0.010
0.005
0.000
Unnormalised Posterior
5.158256e-37
3.807284e-37
2.456312e-37
1.105341e-37
-2.456312e-38
value
mu
variable
Likelihood
Prior
Unnormalised Posterior

```
## **Part 3: The Bayesian learning**

#Number of accidents
k = c( 25, 20, 23, 27)

# After 4 days thus on day 5, final Posterior distribution will be -
post_final = rgamma(1000,40+sum(k),3)

# Histogram plot of prediction of post_final
hist(post_final)
```




```
# predicted accidents on day 5
mean(post_final)
```

```
## [1] 44.98308
```

```
#expectaion is lambda /r for gamma distribution~Gamma(lambda,r)
(40+sum(k))/(3)
```

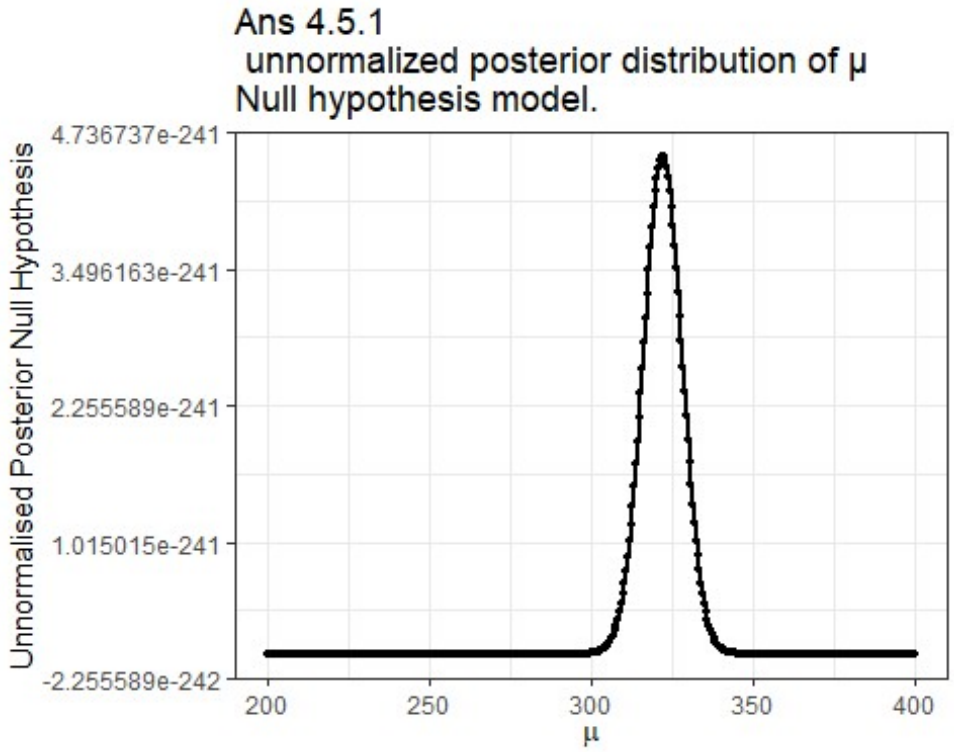
```
## [1] 45
```

```
## **Part 4: Model building in the Bayesian framework**

library(truncnorm)
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

sigma = 60
mu = seq(from =100,to =600, length = 1000)

# NULL Hypothesis Model

delta_null = 0

dat_null =  data.frame(mu = mu,sigma = sigma,delta_null=delta_null)
# likelihoods of words and non words

dat_null$lkl_w = rep(NA,length(mu))
dat_null$lkl_nw = rep(NA,length(mu))

for(i in 1:length(mu))
{
  dat_null$lkl_w[i] = prod(dnorm(dat$Tw,mean = mu[i],sd = sigma))
  dat_null$lkl_nw[i] = prod(dnorm(dat$Tnw,mean = mu[i]+delta_null,sd =
sigma))

}
# now priors
dat_null$prior_mu = dnorm(mu,300,50)
#since for null hypothesis, delta = 0;thus prior or probability of this delta
= 1
dat_null$prior_delta =1

#posterior of Null Hypothesis
dat_null$post_unnorm =
(dat_null$lkl_w)*(dat_null$lkl_nw)*(dat_null$prior_mu)*(dat_null$prior_delta)

##Ans 4.5.1
```

```r
ggplot(dat_null,aes(mu,post_unnorm))+
  geom_point(size =1)+
  geom_line(size =1)+
  theme_bw()+xlab(expression(mu))+ylab("Unnormalised Posterior Null
Hypothesis")+
  scale_x_continuous(limits=c(200,400))+
  labs(title ="Ans 4.5.1\n unnormalized posterior distribution of μ\nNull
hypothesis model.")
```

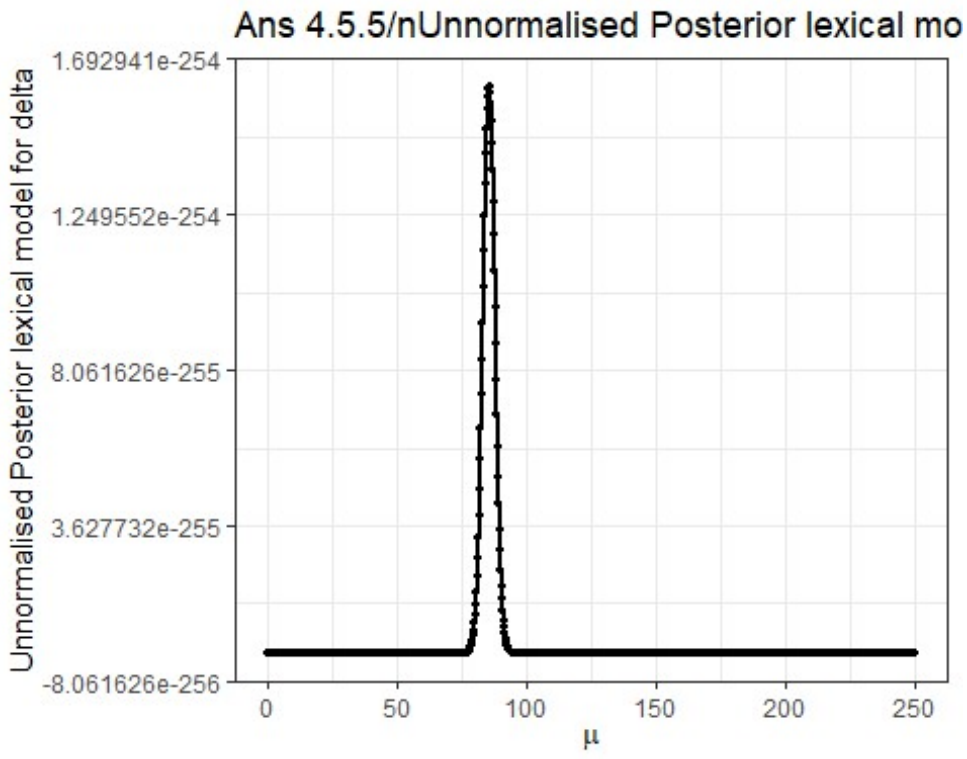
```
## Warning: Removed 600 rows containing missing values or values outside the
scale range
## (`geom_point()`).
```

```
## Warning: Removed 600 rows containing missing values or values outside the
scale range
## (`geom_line()`).
```



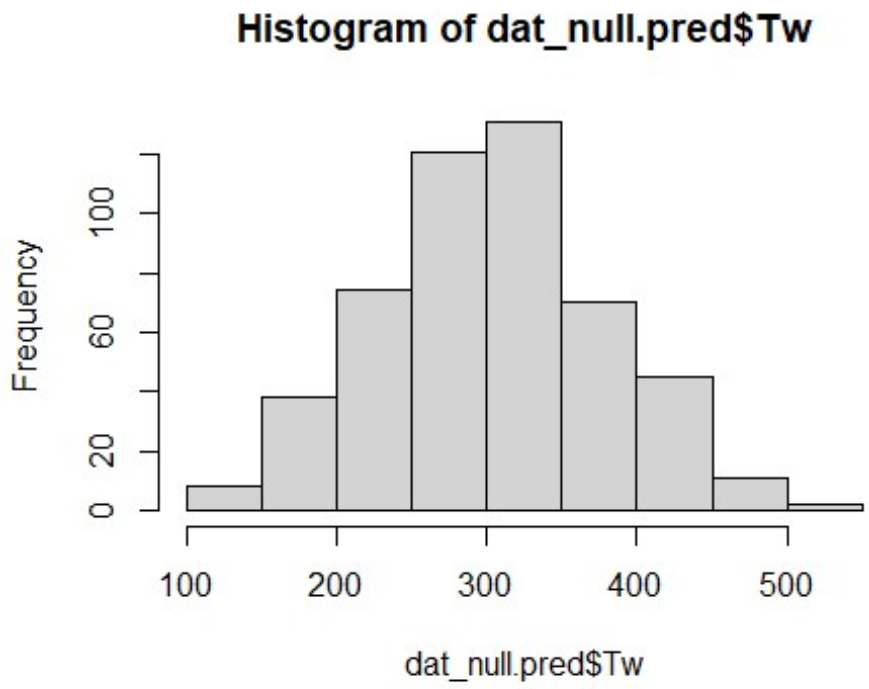

```r
# lexical Access Model

delta_lm = seq(from =0,to=250,length = 1000)
dat_lm = data.frame(mu= mu,sigma = sigma,delta= delta_lm)

# likelihoods of words and non words
dat_lm$lkl_w = rep(NA,length(mu))
dat_lm$lkl_nw = rep(NA,length(mu))
```

```r
for(i in 1 :length(mu))
{
  dat_lm$lkl_w[i] = prod(dnorm(dat$Tw,mean = mu[i],sd = sigma))
  dat_lm$lkl_nw[i] = prod(dnorm(dat$Tnw,mean = (mu[i]+delta_lm[i]),sd =
sigma))
}

#priors
dat_lm$prior_mu = dnorm(mu,300,50)
dat_lm$prior_delta = dtruncnorm(delta_lm,a =0,b= Inf,mean = 0,sd = 50)

#unnormalised posteriors distribution
dat_lm$post_unnorm =
(dat_lm$lkl_w)*(dat_lm$lkl_nw)*(dat_lm$prior_mu)*(dat_lm$prior_delta)

##Ans 4.5.5

ggplot(dat_lm,aes(delta,post_unnorm))+
  geom_point(size =1)+
  geom_line(size =1)+
  theme_bw()+xlab(expression(mu))+ylab("Unnormalised Posterior lexical model
for delta")+
  labs(title = "Ans 4.5.5/nUnnormalised Posterior lexical model for delta")
```




```r
##Ans 4.5.2
```

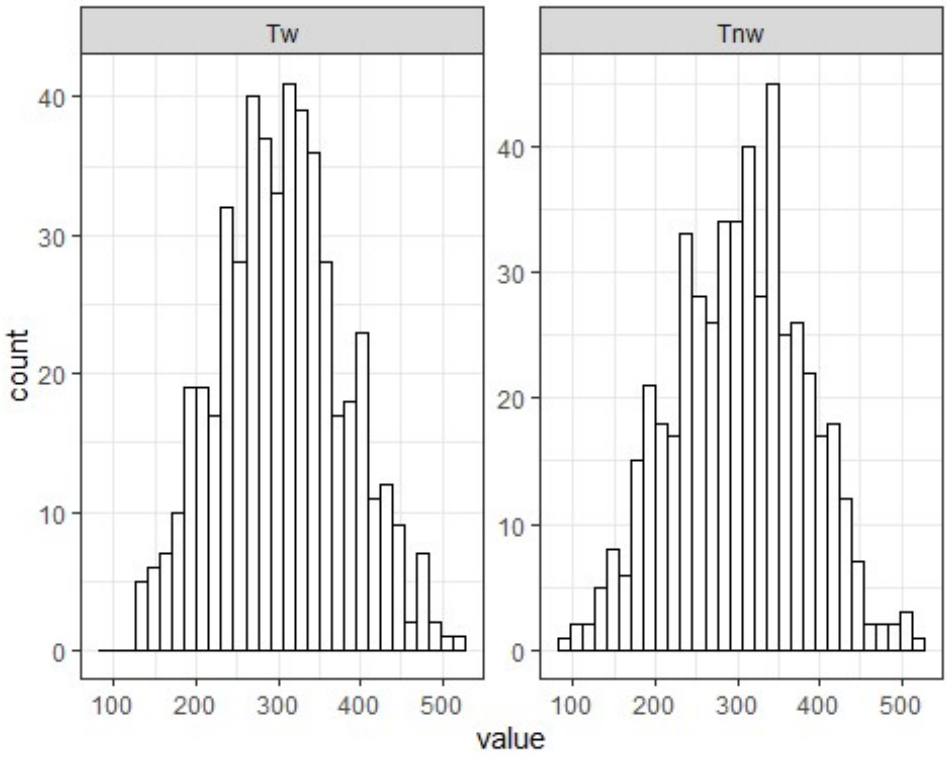
```r
#generating Prior predictions for both the model
##prior predictions from the NULL Hypothesis Model

#creating sample of mu,sigma and delta

sample_mu = rnorm(500,300,50) #we will keep this sample of mu values same for
both the models

#predictions of the null hypo. prior distribution by
#creating a dataframe for it
dat_null.pred = data.frame(mu = sample_mu)
dat_null.pred$Tw = rep(NA,length(sample_mu))
dat_null.pred$Tnw = rep(NA,length(sample_mu))

for(i in 1:length(sample_mu))
{
  dat_null.pred$Tw[i] = rnorm(1,mean  = dat_null.pred$mu[i],sd = sigma)
  dat_null.pred$Tnw[i] = rnorm(1,mean  = dat_null.pred$mu[i],sd = sigma)
}
hist(dat_null.pred$Tw)
```




```r
#plotting together the prior predictions of Null hypo. model for Tw and Tnw
library(reshape2)
dat_null.pred.m = melt(dat_null.pred,id = "mu")
ggplot(dat_null.pred.m, aes(x = value, group = variable))+
  geom_histogram(fill="white",color="black")+
```

```
  theme_bw()+
  facet_wrap(~variable,scales="free_y")
```

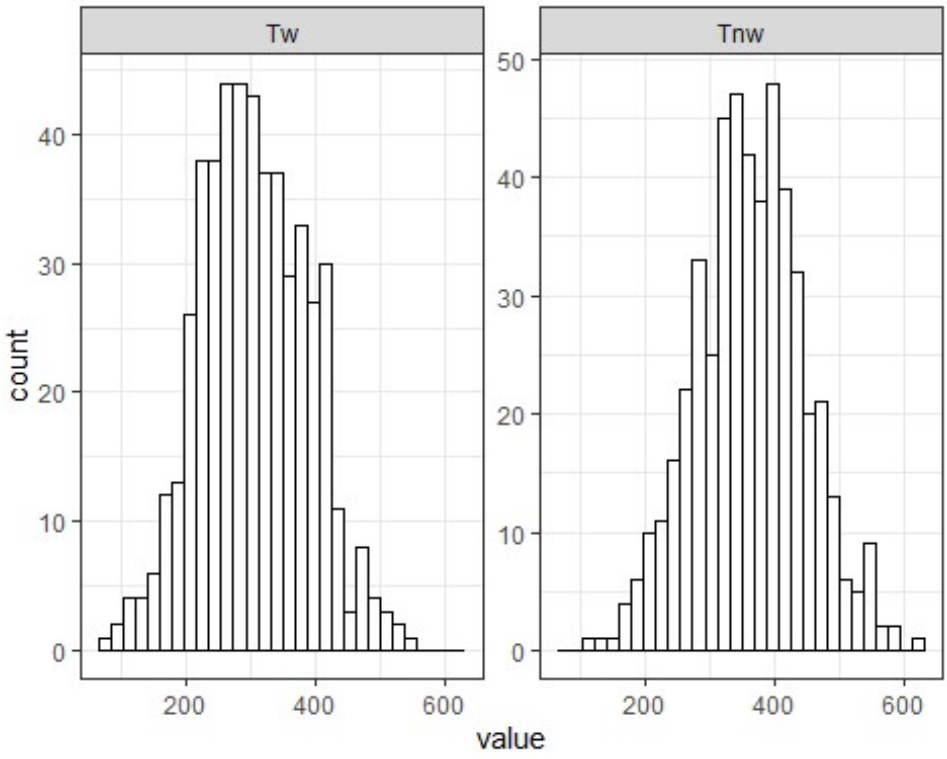
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
## prior predictions from the Lexical Access Model

#sample of delta values
sample_delta_lm = rtruncnorm(500,0,50)

#creating dataframe for it
dat_lm.pred = data.frame(mu = sample_mu, delta = sample_delta_lm)
dat_lm.pred$Tw = rep(NA,length(sample_mu))
dat_lm.pred$Tnw = rep(NA,length(sample_mu))

for(i in 1:length(sample_delta_lm))
{
  dat_lm.pred$Tw[i] = rnorm(1,mean  = dat_lm.pred$mu[i],sd = sigma)
  dat_lm.pred$Tnw[i] = rnorm(1,mean  = (dat_lm.pred$mu[i] +
dat_lm$delta[i]),sd = sigma)
}

#plotting together the prior predictions of Null hypo. model for Tw and Tnw
dat_lm.pred.m = melt(dat_lm.pred,id = c("mu","delta"))
ggplot(dat_lm.pred.m, aes(x = value, group = variable))+
  geom_histogram(fill="white",color="black")+
```

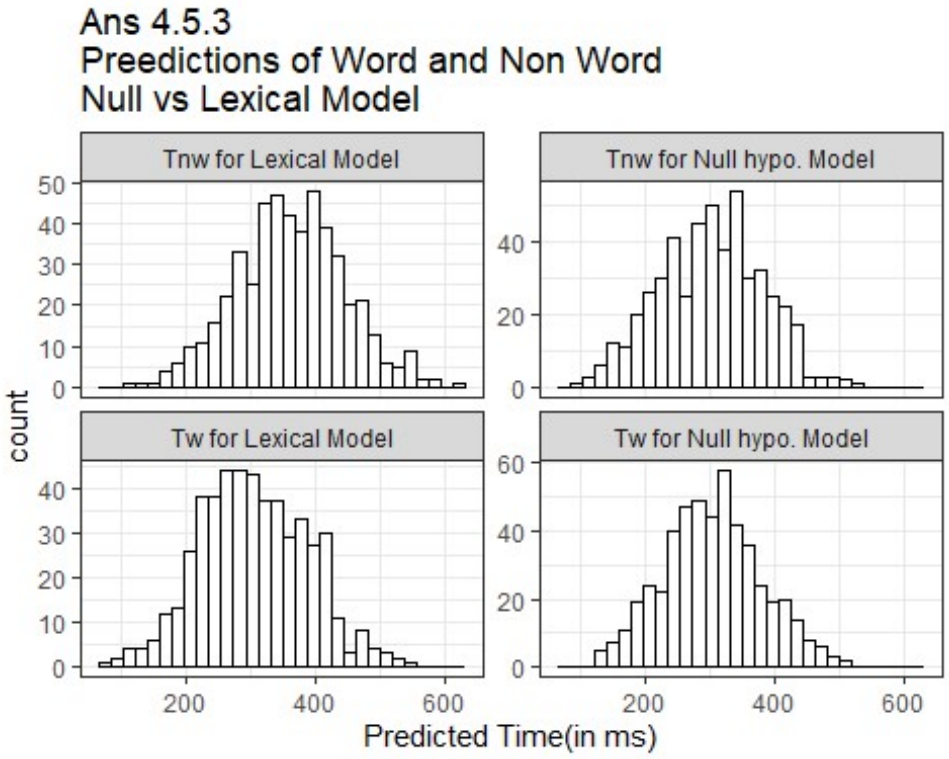
```
  theme_bw()+
  facet_wrap(~variable,scales="free_y",ncol  = 2)
```

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




```
## Ans  4.5.3

#creating a dataframe that stores all prior predictions for bot the models
and
#plotting histogram for all predictions together
dat_pred = data.frame(mu = sample_mu,Tw_null = dat_null.pred$Tw,
                      Tnw_null = dat_null.pred$Tnw,
                      Tw_lm = dat_lm.pred$Tw,Tnw_lm = dat_lm.pred$Tnw)

dat_pred.m = melt(dat_pred,id = "mu")
dat_pred.m$variable = ifelse(dat_pred.m$variable=="Tw_null","Tw for Null
hypo. Model",
                             ifelse(dat_pred.m$variable=="Tnw_null","Tnw for
Null hypo. Model",
                                    ifelse(dat_pred.m$variable=="Tw_lm",
                                           "Tw for Lexical Model","Tnw for
Lexical Model")))

ggplot(dat_pred.m,aes(x = value,group = variable))+
  geom_histogram(fill = "white",color = "black")+
  theme_bw()+
  facet_wrap(~variable,scales = "free_y")+
```

```
  xlab("Predicted Time(in ms)")+
  labs(title ="Ans 4.5.3\nPreedictions of Word and Non Word\nNull vs Lexical
Model")

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

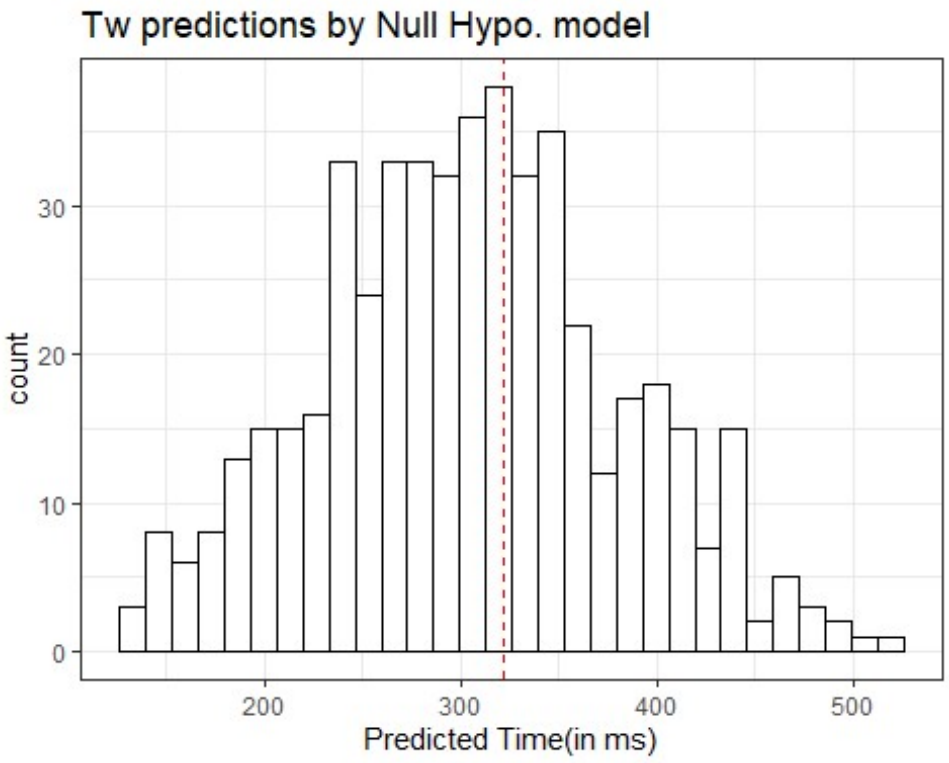



```
##Ans 4.5.4
mean(dat_pred$Tw_null)

## [1] 304.073

mean(dat_pred$Tw_lm)

## [1] 304.685

mean(dat$Tw)

## [1] 321.3746

mean(dat_pred$Tnw_null)

## [1] 301.5777

mean(dat_pred$Tnw_lm)

## [1] 362

mean(dat$Tnw)
```

```
## [1] 323.2388
```

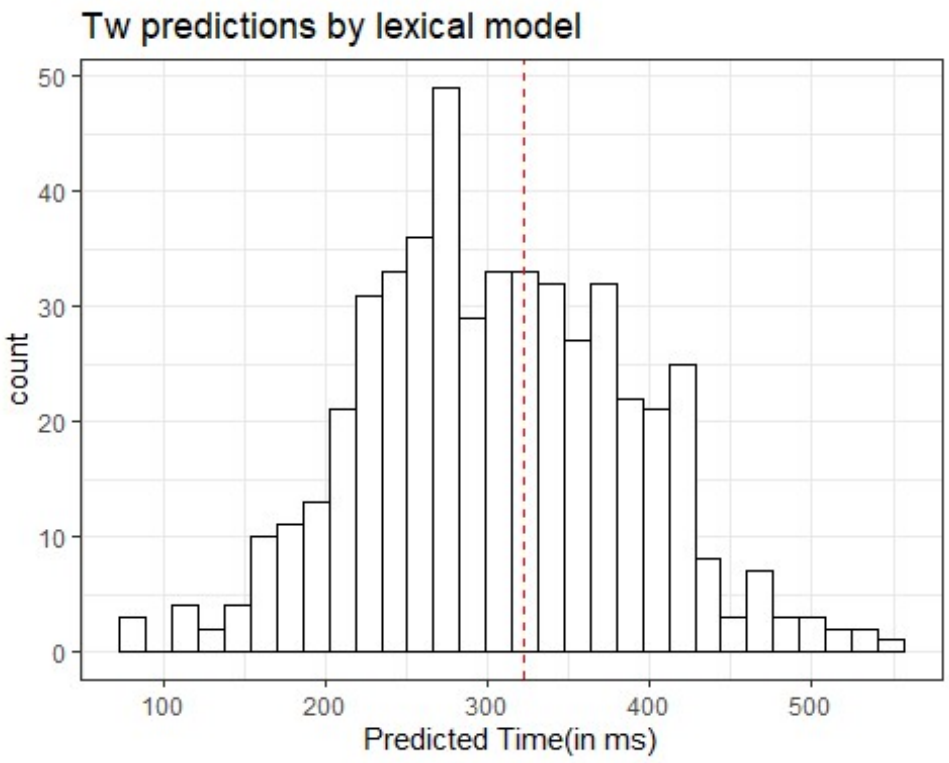
```
#as can be seen clearly that the of predicted  value for words and non words
is close for
#the lexical model while the null hypotheis model is far behind the observed
mean

# plots for words -----

ggplot(dat_pred,aes(x = Tw_null))+
  geom_histogram(fill = "white",color = "black")+
  geom_vline(xintercept = mean(dat$Tw), color = "red", linetype= "dashed")+
  theme_bw()+
  labs(title = "Tw predictions by Null Hypo. model")+
  xlab("Predicted Time(in ms)")
```

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

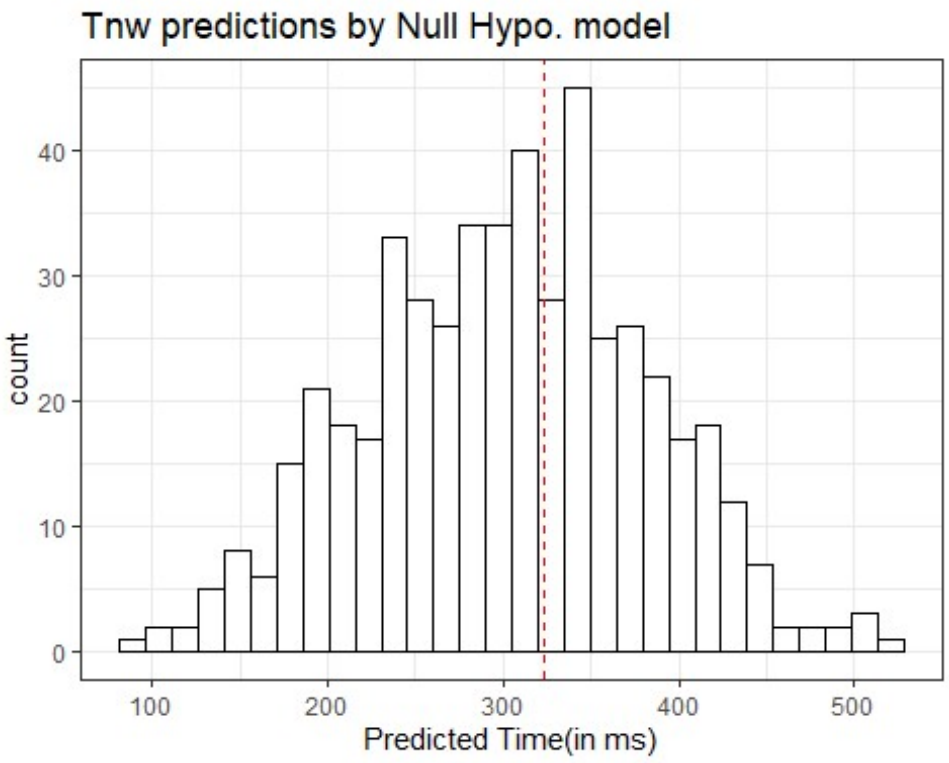



```
ggplot(dat_pred,aes(x = Tw_lm))+
  geom_histogram(fill = "white",color = "black")+
  geom_vline(xintercept = mean(dat$Tw), color = "red", linetype= "dashed")+
  theme_bw()+
  labs(title = "Tw predictions by lexical model")+
  xlab("Predicted Time(in ms)")
```

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

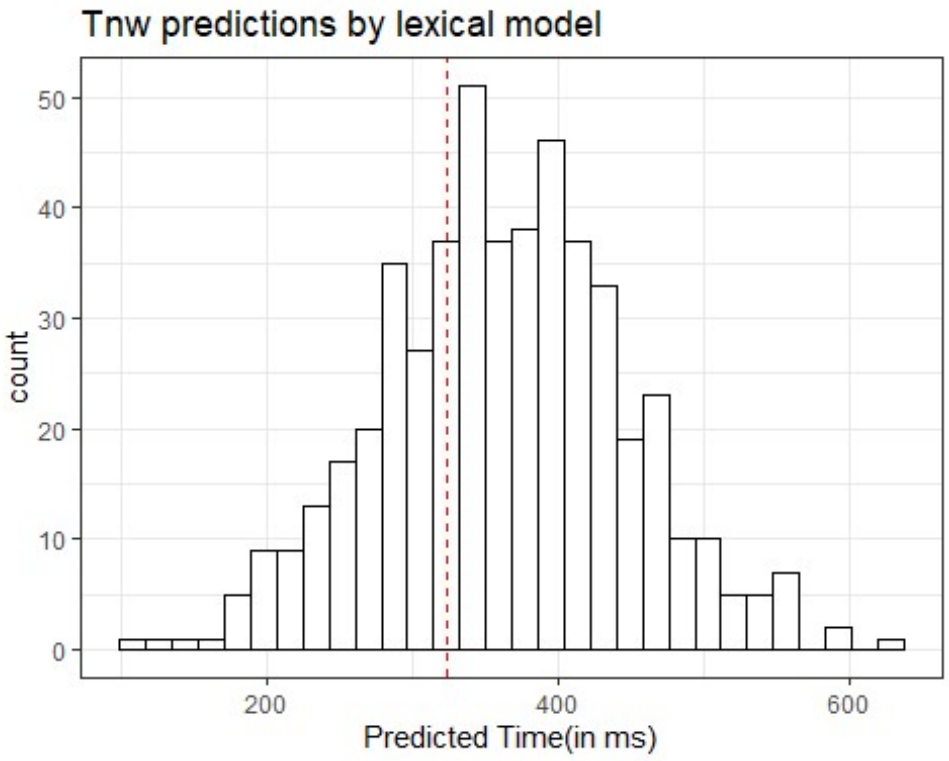



```
# plots for non words -----

ggplot(dat_pred,aes(x = Tnw_null))+
  geom_histogram(fill = "white",color = "black")+
  geom_vline(xintercept = mean(dat$Tnw), color = "red", linetype= "dashed")+
  theme_bw()+
  labs(title = "Tnw predictions by Null Hypo. model")+
  xlab("Predicted Time(in ms)")

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




```
ggplot(dat_pred,aes(x = Tnw_lm))+
  geom_histogram(fill = "white",color = "black")+
  geom_vline(xintercept = mean(dat$Tnw), color = "red", linetype= "dashed")+
  theme_bw()+
  labs(title = "Tnw predictions by lexical model")+
  xlab("Predicted Time(in ms)")
```

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


Tnw predictions by lexical model
count
50
40
30
20
10
0
200
400
600
Predicted Time(in ms)