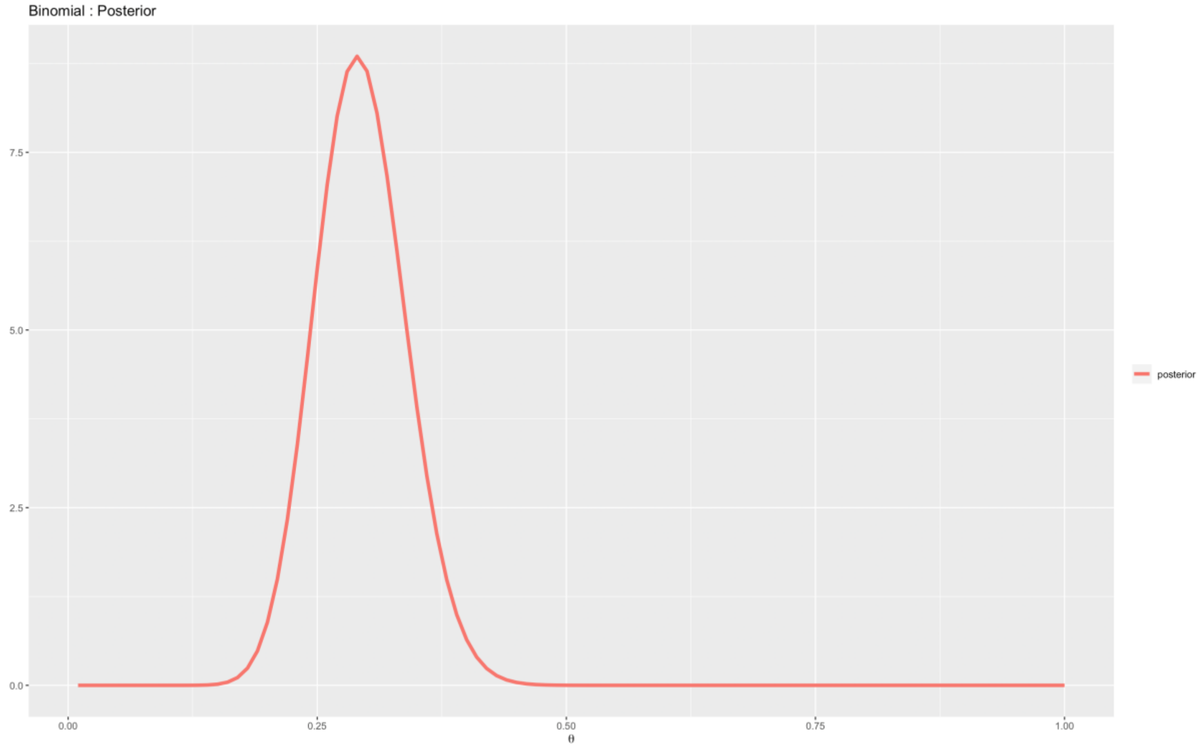
**Assignment 3d**

**Bayes Estimation Outputs**

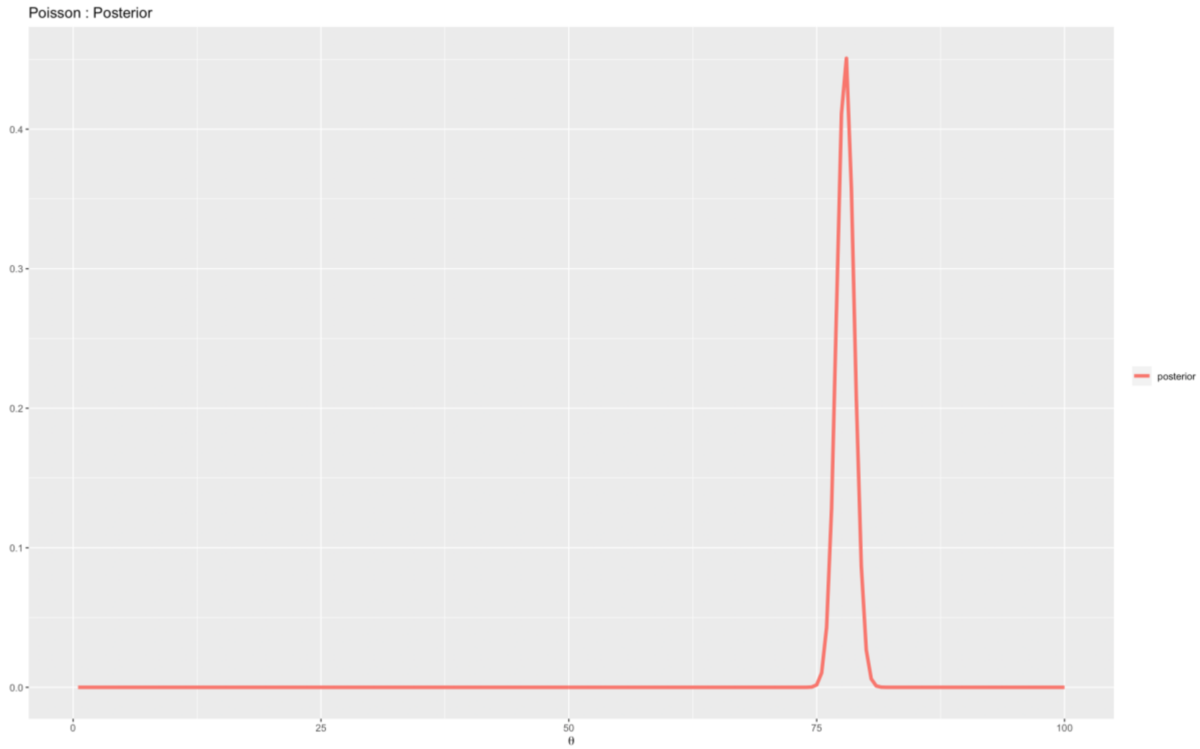
**Binomial:**

```
> #Binomial, #(binom data, dist name, (alpha, beta))
> bayes.estimate(rbinom(100,1,0.3), "Binomial", c(1, 1))
Estimate p: 0.29
```




**Poisson:**

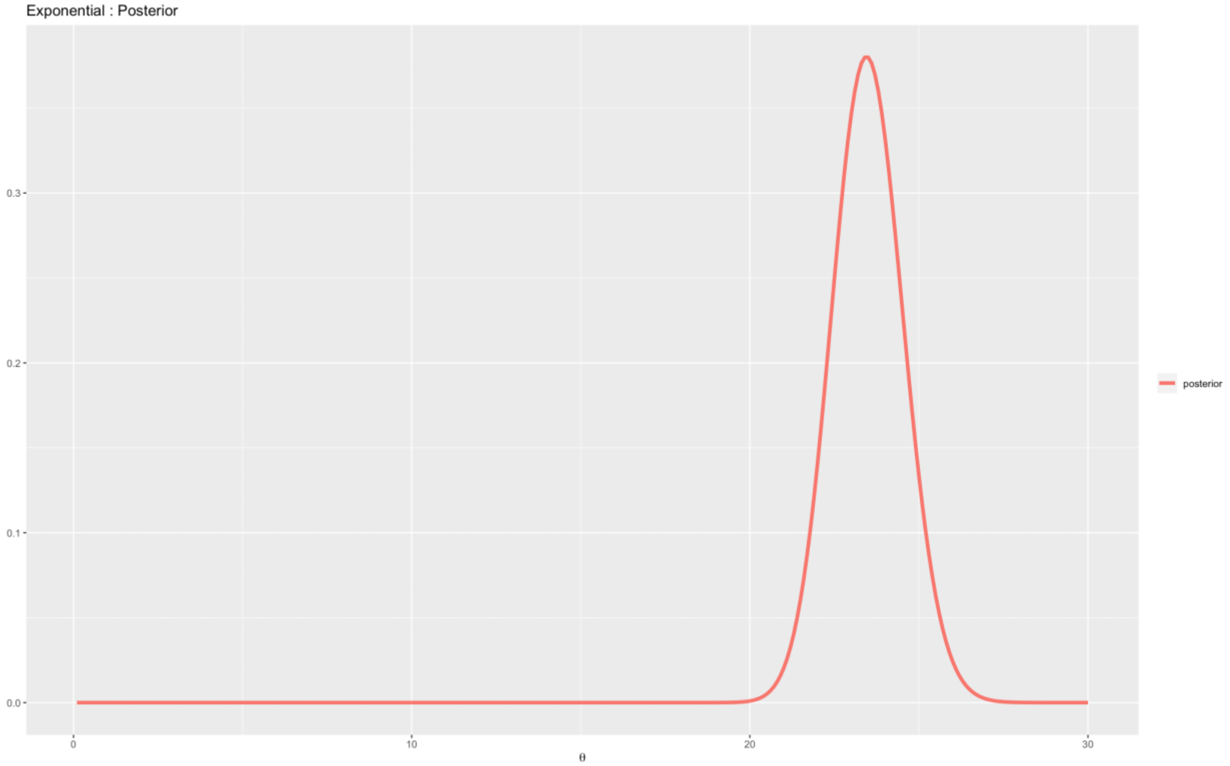
```
> #Poisson, #(Poisson Data, dist name, (alpha, beta))
> bayes.estimate(rpois(100,78), "Poisson", c(1, 1))
Estimate lambda : 78
```



**Exponential:**

> #Exponential, #(exp data, (alpha, beta))
> bayes.estimate(rexp(500,24), "exponential", c(1, 1))

Estimate lambda : 23.5



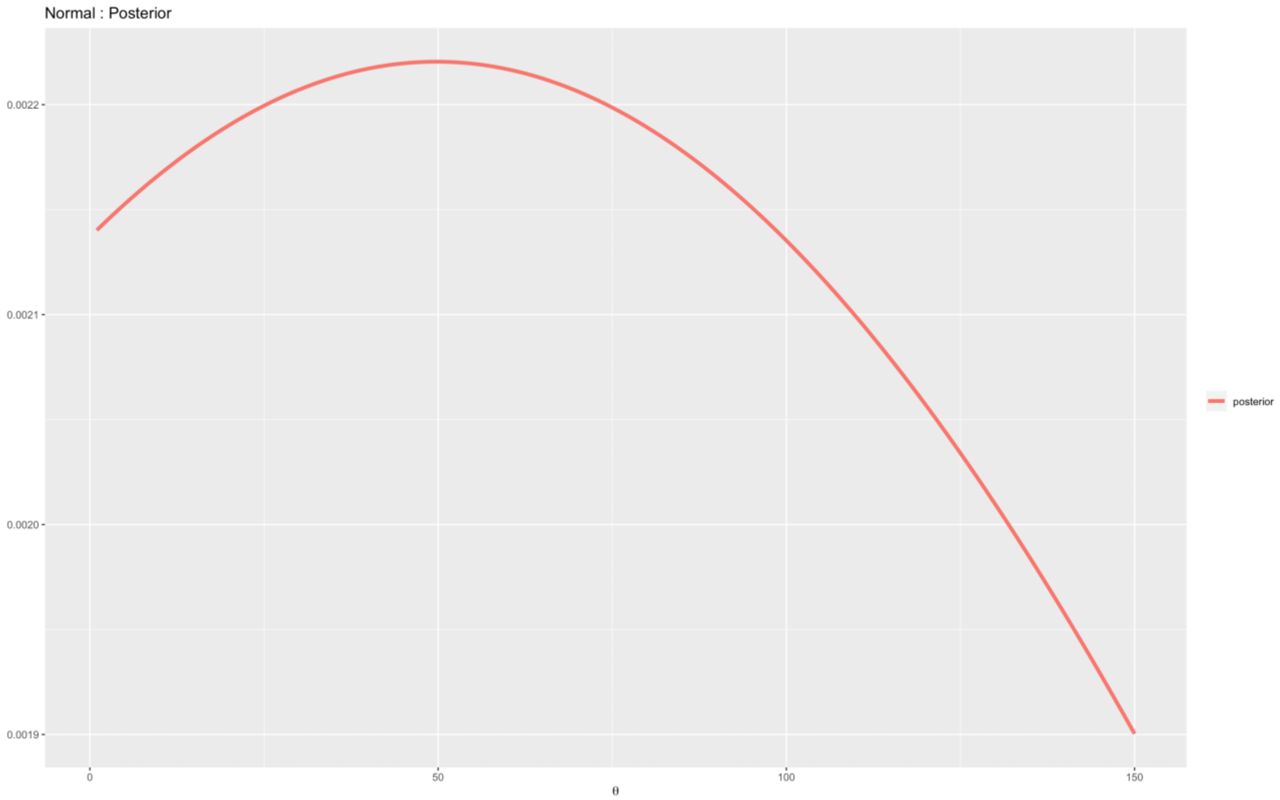

**Normal (Both Unknown) :**

> #normal: both unknown, Normal-Inverse-Gamma, x~N(mu, v/lambda)
> #(normal data, dist name, (mu, lambda, alpha, beta))
> bayes.estimate(rnorm(100,50,2), "normal", c(45, 3, 1, 1))

v = 5.3 , lambda = 3

Estimated taou = 5.3 / 3 = 1.766667

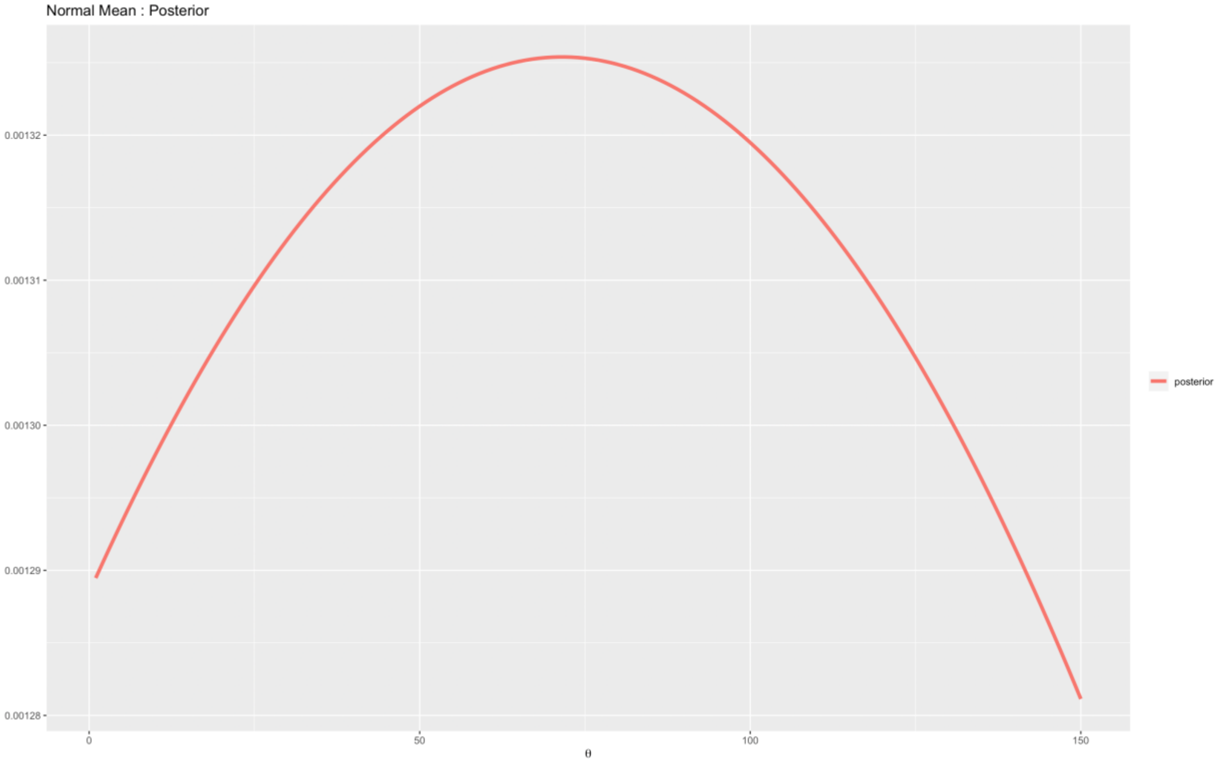
Estimated mu = 50



**Normal (Known Taou, Unknown Mean):**

> #normal: known taou, unknown mean ---> #(normal data, dist name, (mu, taou, known taou))
> bayes.estimate(rnorm(100,72,3), "normal mean", c(66, 1, 3))

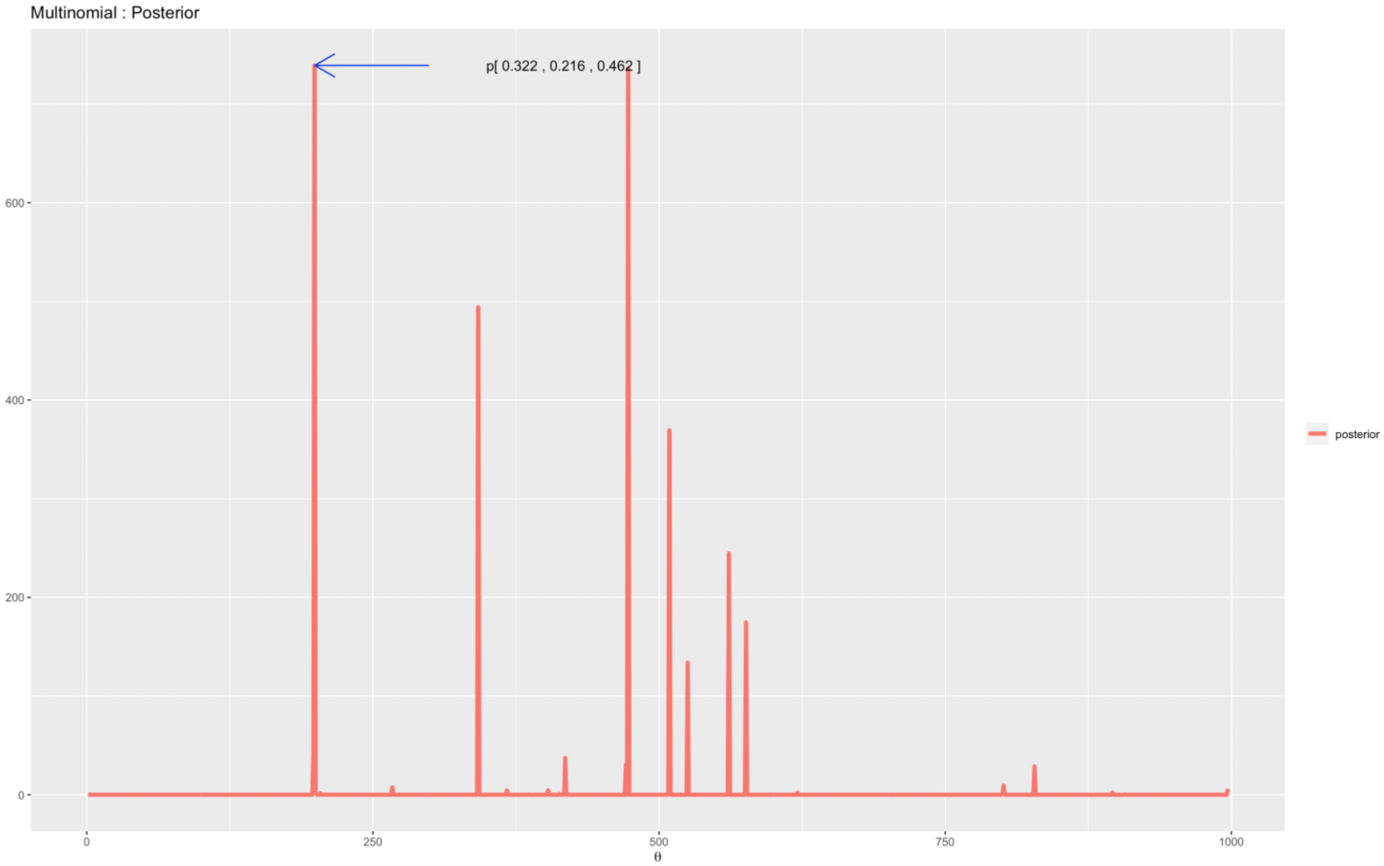
Estimated mu = 72




**Multinomial:**

> #Multinomial, #(multinom data, dist name, alpha vector)
> bayes.estimate(rmultinom(1000, 1, c(0.4, 0.3, 0.5)), "Multinomial", c(1, 1, 1))

Estimated p values : [ 0.322, 0.216, 0.462 ]



**Code:**

```
library(DirichletReg)
library(tidyverse)
library(ggplot2)
library(latex2exp)

# function to initCap input distribution name
initCap <- function(x) {
  s <- strsplit(x, " ")[[1]]
  paste(toupper(substring(s, 1,1)), tolower(substring(s, 2)),
      sep="", collapse=" ")
}

# function to return a row that sums to 1
rand.sum <- function(n){
  x <- round(sort(runif(n-1)),3)
  c(x,1) - c(0,x)
}

# function to plot graphs
final.plot <- function(dist, df){
  if(dist=="Multinomial"){
    sub <- subset(df, posterior == max(df$posterior))
    df <- data.frame(theta = c(1:length(df$theta.1)), posterior = df$posterior)
    data_long <- pivot_longer(df, cols = c(posterior))
    data_long %>%
      ggplot(aes(x = theta, y = value, color = name)) +
      geom_line(size = 1.5) +
      labs(x = TeX("$\\theta$"),
          y = NULL,
          color = NULL,
          title = paste(dist,": Posterior")) +
      annotate("segment",
            x=df[which(df$posterior == max(df$posterior)), "theta"]+100,
            xend=df[which(df$posterior == max(df$posterior)), "theta"]+.2,
            y= df[which(df$posterior == max(df$posterior)), "posterior"],
            yend= df[which(df$posterior == max(df$posterior)), "posterior"],
            arrow=arrow(), color = "blue") +
     annotate(geom = "text", x = df[which(df$posterior == max(df$posterior)), "theta"]+150, y =
df[which(df$posterior == max(df$posterior)), "posterior"], label =
paste("p[",sub$theta.1,",",sub$theta.2,",",sub$theta.3,"]"), hjust = "left")
  }
  else{
    data_long <- pivot_longer(df, cols = c(posterior))
    data_long %>%
      ggplot(aes(x = theta, y = value, color = name)) +
      geom_line(size = 1.5) +
      labs(x = TeX("$\\theta$"),
          y = NULL,
          color = NULL,
          title = paste(dist,": Posterior"))
```

```
  }

}

# function to calculate estimates
bayes.estimate <- function(data, dist, params){
  dist = initCap(dist)
  if (dist == "Binomial"){
    alpha = params[1]
    beta = params[2]
    theta <- seq(.01, 1, .01)
    prior <- dbeta(theta, alpha, beta)
    posterior <- dbeta(theta, alpha + sum(data), beta + length(data) - sum(data))
    df <- data.frame(theta = theta, prior = prior, posterior = posterior)
    sub <- subset(df, posterior == max(df$posterior))
    cat("Estimate p:",sub$theta,"\n")
  }
  if (dist == "Poisson"){
    alpha = params[1]
    beta = params[2]
    theta <- seq(0.5, 100, 0.5)
    prior <- dgamma(theta, alpha, beta)
    posterior <- dgamma(theta, alpha + sum(data), beta + length(data))
    df <- data.frame(theta = theta, prior = prior, posterior = posterior)
    sub <- subset(df, posterior == max(df$posterior))
    cat("Estimate lambda :",sub$theta,"\n")
  }
  if (dist == "Exponential"){
    alpha = params[1]
    beta = params[2]
    theta <- seq(.1, 30, .1)
    prior <- dgamma(theta, alpha, beta)
    posterior <- dgamma(theta, alpha + length(data), beta + sum(data))
    df <- data.frame(theta = theta, prior = prior, posterior = posterior)
    sub <- subset(df, posterior == max(df$posterior))
    cat("Estimate lambda :",sub$theta,"\n")
  }
  if (dist == "Normal"){
    #Normal-inverse-gamma distribution
    #X~N(mu,v/lambda)
    #Both Unknown, processing first for taou using gamma prior
    mu = params[1]
    l = params[2]
    alpha = params[3]
    beta = params[4]
    theta = seq(0, 20, .1)
    prior <- dgamma(theta, alpha, beta)
    a <- vector(mode = "list", length = length(data))
    for (i in 1:length(data)){
      a[i] = (data[i] - mean(data))^2
    }
    a <- as.numeric(unlist(a))
```

```
    n = length(data)
    #Inverse Gamma: 1/x ~ Gamma(new alpha, new beta)
    posterior <- dgamma(1/theta, alpha + n/2, beta + (sum(a)/2) + ((l*n*(mean(data)-mu)^2)/(2*(l+n))))
    df <- data.frame(theta = theta, prior = prior, posterior = posterior)
    sub <- subset(df, posterior == max(df$posterior))
    r = sub$theta/l
    cat("v =",sub$theta,"\n")
    cat("lambda =",l,",","\nEstimated taou =",sub$theta,"/",l,"=",r,"\n")
    plot(theta,posterior,xlab=TeX("$\\theta$"),ylab="density", type="l")
    #calculation for mean using derived value
    theta = seq(1,150,1)
    prior <- dnorm(theta, mu, sub$theta/l)
    posterior <- dnorm(theta, ((l*mu) + (n*r*mean(data)))/(l + (n*r)), l + n*r)
    df <- data.frame(theta = theta, prior = prior, posterior = posterior)
    sub <- subset(df, posterior == max(df$posterior))
    cat("Estimated mu =",sub$theta,"\n")
  }
  if (dist == "Normal Mean"){
    #known Taou, unknown Mean
    mu = params[1]
    taou = params[2]
    r = params[3]
    n = length(data)
    theta = seq(1,150,1)
    prior <- dnorm(theta, mu, taou)
    posterior <- dnorm(theta, ((taou*mu) + (n*r*mean(data)))/(taou + (n*r)), taou + n*r)
    df <- data.frame(theta = theta, prior = prior, posterior = posterior)
    sub <- subset(df, posterior == max(df$posterior))
    cat("Estimated mu =",sub$theta,"\n")
  }
  if (dist == "Multinomial"){
    alpha = params
    theta <- t(replicate(1000,rand.sum(3)))
    theta <- unique(theta)
    prior <- ddirichlet(theta, alpha)
    x <- c(sum(data[1,]), sum(data[2,]), sum(data[3,]))
    posterior <- ddirichlet(theta, alpha + x)
    df <- data.frame(theta = theta, prior = prior, posterior = posterior)
    sub <- subset(df, posterior == max(df$posterior))
    cat("Estimated p values : [",sub$theta.1, sub$theta.2, sub$theta.3,"] \n")
  }

  final.plot(dist, df)
}

# Binomial, #(binom data, dist name, (alpha, beta))
bayes.estimate(rbinom(100,1,0.3), "Binomial", c(1, 1))

# Poisson, #(Poisson Data, dist name, (alpha, beta))
bayes.estimate(rpois(100,78), "Poisson", c(1, 1))
```

```r
# Exponential, #(exp data, (alpha, beta))
bayes.estimate(rexp(500,24), "exponential", c(1, 1))

# normal: both unknown, Normal-Inverse-Gamma, x~N(mu, v/lambda)
# (normal data, dist name, (mu, lambda, alpha, beta))
bayes.estimate(rnorm(100,50,2), "normal", c(45, 3, 1, 1))

# normal: known taou, unknown mean ---> #(normal data, dist name, (mu, taou, known taou))
bayes.estimate(rnorm(100,72,3), "normal mean", c(66, 1, 3))

# Multinomial, #(multinom data, dist name, alpha vector)
bayes.estimate(rmultinom(1000, 1, c(0.4, 0.3, 0.5)), "Multinomial", c(1, 1, 1))
```