Bayes Estimate

Prepared by: Josh Levine (jl2108), Harsh Patel (hkp49), Jaini Patel (jp1891), Yifan Liao (yl1463) and Aayush Shah (avs93).

Assignment

Construct Bayesian estimates using the conjugate priors for each of the method of moments estimates. Input is data, name, and prior distribution parameters. Output is the posterior distribution for each parameter (plotted) with probability bounds for the parameter (with input confidence level)

What is a Bayesian Estimator?

A Bayesian estimator is an estimator of an unknown parameter Theta that minimizes the expected loss for all observations x of X. In other words, it's a term that estimates your unknown parameter in a way that you lose the least amount of accuracy. A Bayesian estimator is a function of observable random variables, variables you observed in the process of your research.

Located below is the code for goodness of fit test, Method of moments estimator, Maximum likelihood estimator, and Bayes estimates of posterior distributions

```
# Baysian Estimate for distributions using conjugate priors

library(actuar) #for rpareto function

# Compute 1st moment for the sample around origin to get Sample_Mean
firstMoment <- function(vec){ sum =
  sum(vec,na.rm=TRUE) return
  (sum/length(vec))
}

# Compute 2nd moment for the sample around the sample mean to get Sample_Variance
secondMoment <- function(vec){ mu =
  firstMoment(vec)
  square <- sum((vec - mu)^2,na.rm=TRUE)
  return (square/length(vec))
}

binomial_mom <- function(vec){ mu =
  firstMoment(vec)
  var = secondMoment(vec)

  p_hat =1      - (var/mu)
  n_hat =mu /p_hat

  list(n_hat =n_hat,p_hat =p_hat)
}

poisson_mom <- function(vec){
  lambda_hat = firstMoment(vec)
  return(lambda_hat)
```

```
}

exponential_mom <- function(vec){
   beta_hat =1 /firstMoment(vec)
   return(beta_hat)
}

geometric_mom <- function(vec){ p_hat =1
   /firstMoment(vec) - 1 return(p_hat)
}

uniform_mom <- function(vec){ mu =
   firstMoment(vec)
   var = secondMoment(vec) hold
   =var *sqrt(3)

   a_hat =mu      - hold
   b_hat =mu      + hold

   list(a_hat =a_hat,b_hat =b_hat)
}

normal_mom <- function(vec){ mu_hat =
   firstMoment(vec)
   sd_hat = sqrt(secondMoment(vec))

   list(mu_hat =mu_hat,sd_hat =sd_hat)
}

binomial_mle <-function(input_data,n)
{
   binomial_mle <- mean(input_data) return(binomial_mle)
}

poisson_mle <- function(input_data)
{
   lambda_value_mle <- mean(input_data)
   return(lambda_value_mle)
}

exponential_mle <- function(input_data)
{
   beta_mle <-1 /mean(input_data)
   return(beta_mle)
}

geometric_mle <- function(input_data)
{
   geometric_mle <-(1.0 /(mean(input_data)+1))
   return(geometric_mle)
}
```

```r
uniform_mle <-function(input_data)
{
   a_mle     <-     min(input_data)
   b_mle     <-     max(input_data)
   return(c(a_mle,b_mle))
}

normal_mle <-function(input_data)
{
   n <- length(input_data) mean_mle <-
   mean(input_data)
   variance_mle <- sum((input_data-mean_mle)^2)/n
   return(c(mean_mle,sqrt(variance_mle)))
}


bayes_estimate <- function(distribution, sample,nboot=1000){

   cat("\n")
   print(paste("-------", distribution," -----------------------------"))
   n = length(sample) if(distribution ==
   "Binomial"){
      p =0.4
      # Conjugate prior - Beta(alpha, beta)
      prior_alpha =1
      prior_beta =1

      # Posterior - Beta(alpha + n*mean(X), beta + n - n*mean(X))
      posterior_alpha =prior_alpha              +sum (sample)
      posterior_beta =prior_beta              +length (sample) -sum (sample)

      # Bayes estimate of p - Mean of posterior distribution
      estimate =posterior_alpha / (posterior_alpha + posterior_beta)

      print("Paramters of posterior beta distribution: ")
      print(c(posterior_alpha, posterior_beta)) print(paste("p = ", p))
      print(paste("Bayes estimate = ", estimate))
      cat("\n")

      # Dentisy
      posterior_sample <- rbeta(n, posterior_alpha, posterior_beta)
      plot(density(posterior_sample))

      #MOM
      n_ =1000
      l = binomial_mom(sample)
      print("Method of Moments:")
      print(paste0("Population Parameters:                    ","n_ = ", n_,"                              p = ", p))              round(l$p_ha
      print(paste0("Estimated parameters: ","n_hat = ", round(l$n_hat,3) ,"                              p_hat = ", cat("\n")
      #MOM
```

```
    #MLE
    theta_hat = binomial_mle(sample, n)
    print("MLE: ") print(theta_hat)
    cat("\n")
    #MLE

    #Goodness of fit
    q_hat <- qbinom(c(1:n)/(n+1), n,theta_hat)

    D0 <- ks.test(sample, q_hat)$statistic Dvec<-
    NULL

    for(i in 1:nboot){
        x_star <- rbinom(1000,1,0.4) theta_hat_star <-
        binomial_mle(x_star, n)

        q_hat_star <- qbinom(c(1:n)/(n+1), n, theta_hat_star) D_star <-
        ks.test(x_star, q_hat_star)$statistic
        Dvec <- c(Dvec, D_star)
    }
    p_value <- sum(Dvec > D0)/nboot print("Goodness of fit
    p-value: ") print(p_value)
    cat("\n")
    #Goodness of fit
}

elseif (distribution == "Poisson"){
    # Conjugate prior - Gamma(alpha, beta)
    prior_alpha =1
    prior_beta =1

    # Posterior - Gamma(alpha + n*mean(X), beta +n)
    posterior_alpha =prior_alpha          +sum (sample)
    posterior_beta =prior_beta            +length (sample)

    # Bayes estimate - Mean of posterior distribution
    estimate =posterior_alpha / posterior_beta

    print("Paramters of posterior gamma distribution: ")
    print(c(posterior_alpha, posterior_beta)) print(paste("Bayes
    estimate = ", estimate)) cat("\n")

    # Dentisy
    posterior_sample <- rgamma(n, posterior_alpha, posterior_beta)
    plot(density(posterior_sample))

    #MOM
    lambda =5
    lambda_hat = poisson_mom(sample) print("Method of
    Moments:")
```

```
    print(paste0("Population Parameter:                   ","lambda = ", lambda))
    print(paste0("Estimated parameter:","lambda_hat = ", round(lambda_hat,3))) cat("\n")
    #MOM

    #MLE
    theta_hat = poisson_mle(sample) print("MLE: ")
    print(theta_hat) cat("\n")
    #MLE

    #Goodness of fit
    q_hat <- qpois(c(1:n)/(n+1),theta_hat)

    D0 <- ks.test(sample, q_hat)$statistic Dvec<-
    NULL

    for(i in 1:nboot){
        x_star <- rpois(n,theta_hat) theta_hat_star <-
        poisson_mle(x_star)

        q_hat_star <- qpois(c(1:n)/(n+1), theta_hat_star) D_star <-
        ks.test(x_star, q_hat_star)$statistic Dvec <- c(Dvec, D_star)
    }
    p_value <- sum(Dvec > D0)/nboot print("Goodness of fit
    p-value: ") print(p_value)
    cat("\n")
    #Goodness of fit
}

elseif (distribution == "Exponential"){ # Conjugate
    prior - Gamma(alpha, beta) prior_alpha =1
    prior_beta =1

    # Posterior -           Gamma(alpha + n, beta +n*mean(X))
    posterior_alpha =prior_alpha          +length (sample)
    posterior_beta =prior_beta            +sum (sample)

    # Bayes estimate - Mean of posterior distribution
    estimate =posterior_alpha / posterior_beta

    print("Paramters of posterior gamma distribution: ")
    print(c(posterior_alpha, posterior_beta)) print(paste("Bayes
    estimate = ", estimate)) cat("\n")

    # Dentisy
    posterior_sample <- rgamma(n, posterior_alpha, posterior_beta)
    plot(density(posterior_sample))
```

```
    #MOM
    beta =5
    beta_hat = exponential_mom(sample) print("Method of
    Moments:")
    print(paste0("Population Parameter:                    ","beta = ", beta))
    print(paste0("Estimated parameter:","beta_hat = ", round(beta_hat,3))) cat("\n")
    #MOM

    #MLE
    theta_hat = exponential_mle(sample) print("MLE: ")
    print(theta_hat) cat("\n")
    #MLE

    #Goodness of fit
    q_hat <- qexp(c(1:n)/(n+1),theta_hat)

    D0 <- ks.test(sample, q_hat)$statistic Dvec<-
    NULL

    for(i in 1:nboot){
        x_star <- rexp(n,theta_hat) theta_hat_star <-
        exponential_mle(x_star)

        q_hat_star <- qexp(c(1:n)/(n+1), theta_hat_star) D_star <-
        ks.test(x_star, q_hat_star)$statistic Dvec <- c(Dvec, D_star)
    }
    p_value <- sum(Dvec > D0)/nboot print("Goodness of fit
    p-value: ") print(p_value)
    cat("\n")
    #Goodness of fit
}

elseif (distribution == "Geometric"){
    # Conjugate prior - Beta(alpha, beta)
    prior_alpha =1
    prior_beta =1

    # Posterior - Beta(alpha + n, beta +n*mean(X))
    posterior_alpha =prior_alpha           +length (sample)
    posterior_beta =prior_beta             +sum (sample)

    # Bayes estimate - Mean of posterior distribution
    estimate =posterior_alpha / (posterior_alpha + posterior_beta)

    print("Paramters of posterior beta distribution: ")
    print(c(posterior_alpha, posterior_beta)) print(paste("Bayes
    estimate = ", estimate)) cat("\n")
```

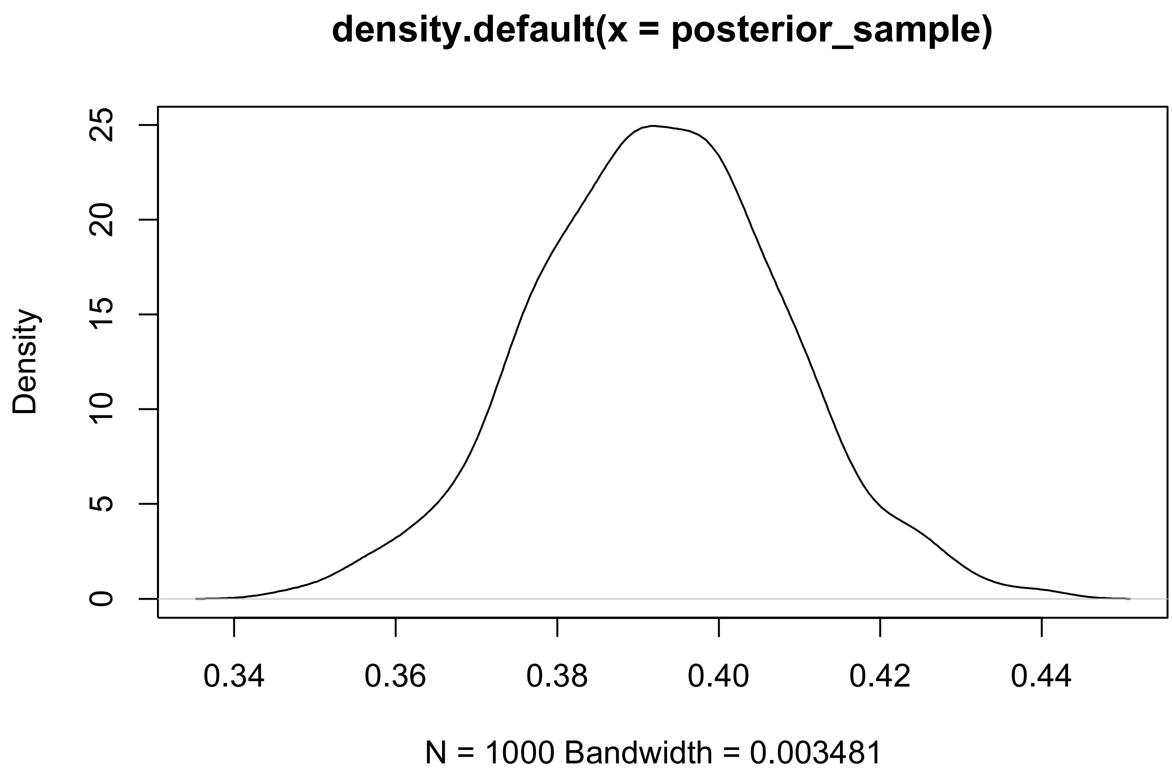
```
    # Dentisy
    posterior_sample <- rbeta(n, posterior_alpha, posterior_beta)
    plot(density(posterior_sample))

    #MOM
    p =0.7
    p_hat = geometric_mom(sample)

    print("Method of Moments:")
    print(paste0("Population Parameter:                    ","p = ", p))
    print(paste0("Estimated parameter:","p_hat = ", round(p_hat,3))) cat("\n")
    #MOM

    #MLE
    theta_hat = geometric_mle(sample) print("MLE:
    ")
    print(theta_hat) cat("\n")
    #MLE

    #Goodness of fit
    q_hat  <-  qgeom(c(1:n)/(n+1),theta_hat)   D0   <-
    ks.test(sample,q_hat)$statistic Dvec<-NULL

    for(i in 1:nboot){
        x_star <- rgeom(n,theta_hat) theta_hat_star <-
        geometric_mle(x_star)

        q_hat_star <- qgeom(c(1:n)/(n+1), theta_hat_star) D_star <-
        ks.test(x_star, q_hat_star)$statistic Dvec <- c(Dvec, D_star)
    }
    hist(Dvec)
    p_value <- sum(Dvec > D0)/nboot print("Goodness of fit
    p-value: ") print(p_value)
    cat("\n")
    #Goodness of fit
}

elseif (distribution == "Uniform"){ #
    Conjugate prior - Pareto prior_v0 =1
    prior_k =1

    # Posterior
    posterior_v0 = max(c(prior_v0,sample))
    posterior_k =prior_k             +length (sample)

    print("Paramters of posterior pareto distribution: ")
    print(c(posterior_v0, posterior_k))
```

```
    cat("\n")

    # Dentisy
    posterior_sample <- rpareto(n, posterior_v0, posterior_k)
    plot(density(posterior_sample))

    #MOM
    a =0
    b =10
    l = uniform_mom(sample)

    print("Method of Moments:")
    print(paste0("Population Parameters:                    ","a = ", a,"                                      b = ", b)) print(paste0("Estimated
    parameters: ","a_hat = ", round(l$a_hat,3),"            b_hat = ",                                      round(l$b_hat cat("\n")
    #MOM

    #MLE
    theta_hat = uniform_mle(sample) print("MLE: ")
    print(theta_hat) cat("\n")
    #MLE

    #Goodness of fit
    q_hat <- qunif(c(1:n)/(n+1), theta_hat[1], theta_hat[2])

    D0 <- ks.test(sample, q_hat)$statistic Dvec<-
    NULL

    for(i in 1:nboot){
        x_star <- runif(n,theta_hat[1],theta_hat[2]) theta_hat_star <-
        uniform_mle(x_star)

        q_hat_star <- qunif(c(1:n)/(n+1), theta_hat_star[1], theta_hat_star[2]) D_star <-
        ks.test(x_star, q_hat_star)$statistic
        Dvec <- c(Dvec, D_star)
    }
    p_value <- sum(Dvec > D0)/nboot print("Goodness of fit
    p-value: ") print(p_value)
    cat("\n")
    #Goodness of fit
}

elseif (distribution == "Normal"){
    # Assuming alpha and beta for the prior distribution to be 1
    r <-1
    tau <-5
    mu <-4
    prior_alpha <-1
    prior_beta <-2
```

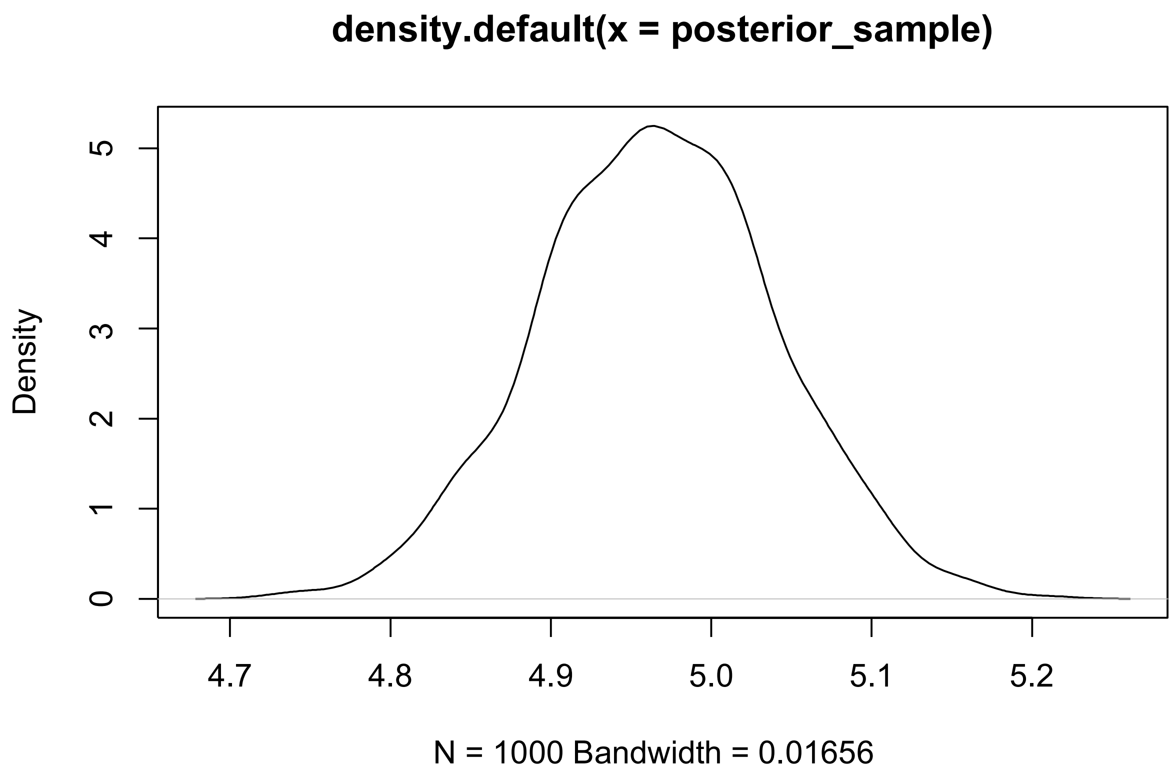
```
# Getting the posterior distribution parameters
M_conditional_distribution_mu <-(tau *mu +length (sample)*mean(sample))/(tau +length (sample))
M_conditional_distribution_precision <-(tau                    +length (sample))*r
print("The parameters of the conditional posterior normal distribution of M when R=r is:")
print(c(M_conditional_distribution_mu, M_conditional_distribution_precision))

R_marginal_distribution_alpha <-prior_alpha                    +length (sample)/2
R_marginal_distribution_beta <-prior_beta                  + 1/2*(sum((sample -mean (sample))**2)) +tau*length(sam print("The
parameters of the marginal posterior gamma distribution of R is:") print(c(R_marginal_distribution_alpha,
R_marginal_distribution_beta))

# Generate the distibutions
conditional_joint_distribution_of_M <- rnorm(n,mean =M_conditional_distribution_mu,1                              /sqrt
marginal_joint_distribution_of_R <- rgamma(n, R_marginal_distribution_alpha,R_marginal_distributio

plot(density(conditional_joint_distribution_of_M))
plot(density(marginal_joint_distribution_of_R))

#MOM
mu =4
sd =20
l = normal_mom(sample)

print("Method of Moments:")
print(paste0("Population Parameters:                   ","mu = ", mu,"                              sd = ", sd)) print(paste0("Estimated
parameters: ","mu_hat = ", round(l$mu_hat,3),"     sd_hat = ",                                    round(l$sd cat("\n")
#MOM

#MLE
theta_hat = normal_mle(sample)
print("MLE: ") print(theta_hat)
cat("\n")
#MLE

#Goodness of fit
q_hat <- qnorm(c(1:n)/(n+1),mean =theta_hat[1],sd =theta_hat[2])

D0 <- ks.test(sample, q_hat)$statistic Dvec<-
NULL

for(i in 1:nboot){
   x_star <- rnorm(n,theta_hat[1],theta_hat[2]) theta_hat_star <-
   normal_mle(x_star)

   q_hat_star <- qnorm(c(1:n)/(n+1),mean =theta_hat_star[1],sd =theta_hat_star[2]) D_star <- ks.test(x_star,
   q_hat_star)$statistic
   Dvec <- c(Dvec, D_star)
}
p_value <- sum(Dvec > D0)/nboot print("Goodness of fit
p-value: ") print(p_value)
```

```r
    cat("\n")
    #Goodness of fit
  }

  print(paste("-------End of ", distribution," --------------------------------- "))
}

#bayes_estimate_wrapper("Binomial")


#################################################################
#### Main wrapper function to run bayesian estimate for all distributions####
#################################################################
main <- function(){

  bayes_estimate("Binomial", rbinom(n =1000,size =1,0.4))
  bayes_estimate("Poisson",        rpois(n =1000,lambda =5))
  bayes_estimate("Exponential",    rexp(n =1000,rate =5))
  bayes_estimate("Geometric",      rgeom(n =1000,0.7))
  bayes_estimate("Uniform", runif(n =1000,min =0,max=10))
  bayes_estimate("Normal", rnorm(n =1000,mean =10,sd=20))

}


main()
```

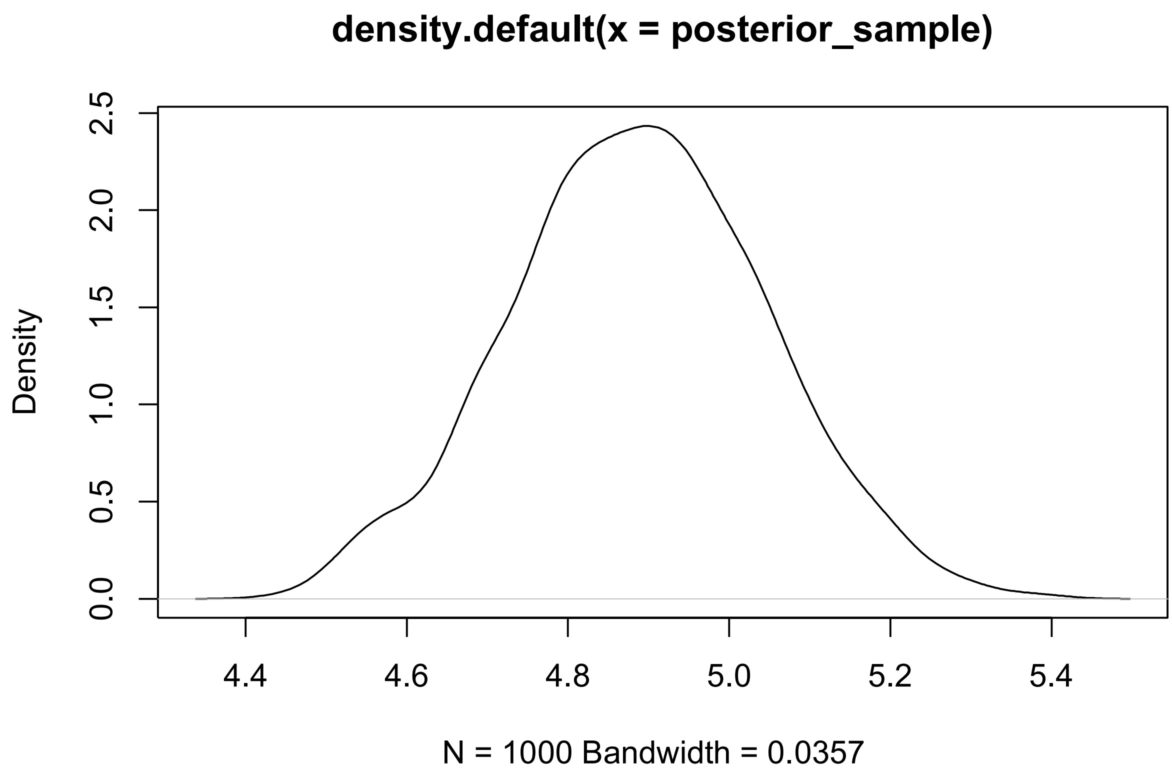
```
##
## [1] "------- Binomial ------------------------- "
## [1] "Paramters of posterior beta distribution: " ## [1] 393 609
## [1] "p =         0.4"
## [1] "Bayes estimate =             0.392215568862275"
```



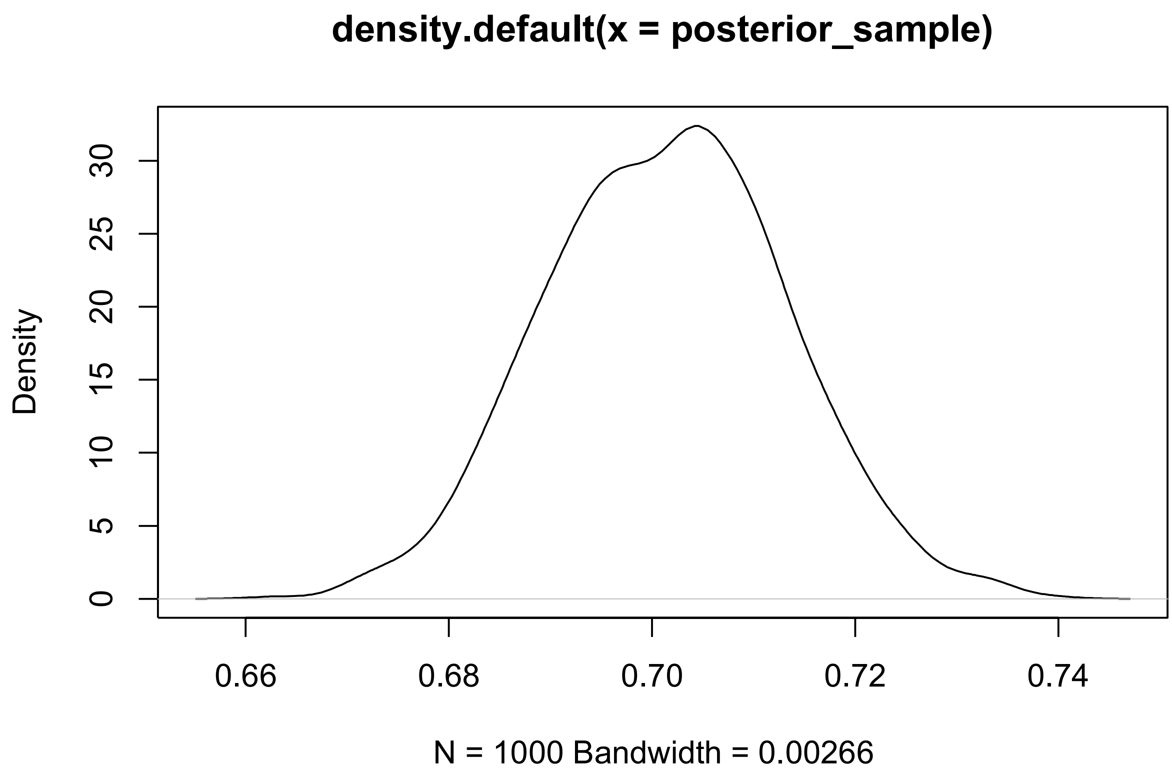

```
## [1] "Method of Moments:"
## [1] "Population Parameters:        n_ = 1000              p = 0.4"
## [1] "Estimated parameters: n_hat = 1              p_hat = 0.392" ##

## [1] "MLE: "
## [1] 0.392
##
## [1] "Goodness of fit p-value: "
## [1] 0
##
## [1] "-------End of      Binomial ---------------"
##
## [1] "------- Poisson----------------------"
## [1] "Paramters of posterior gamma distribution: "
## [1] 4973 1001
## [1] "Bayes estimate =      4.96803196803197"
```



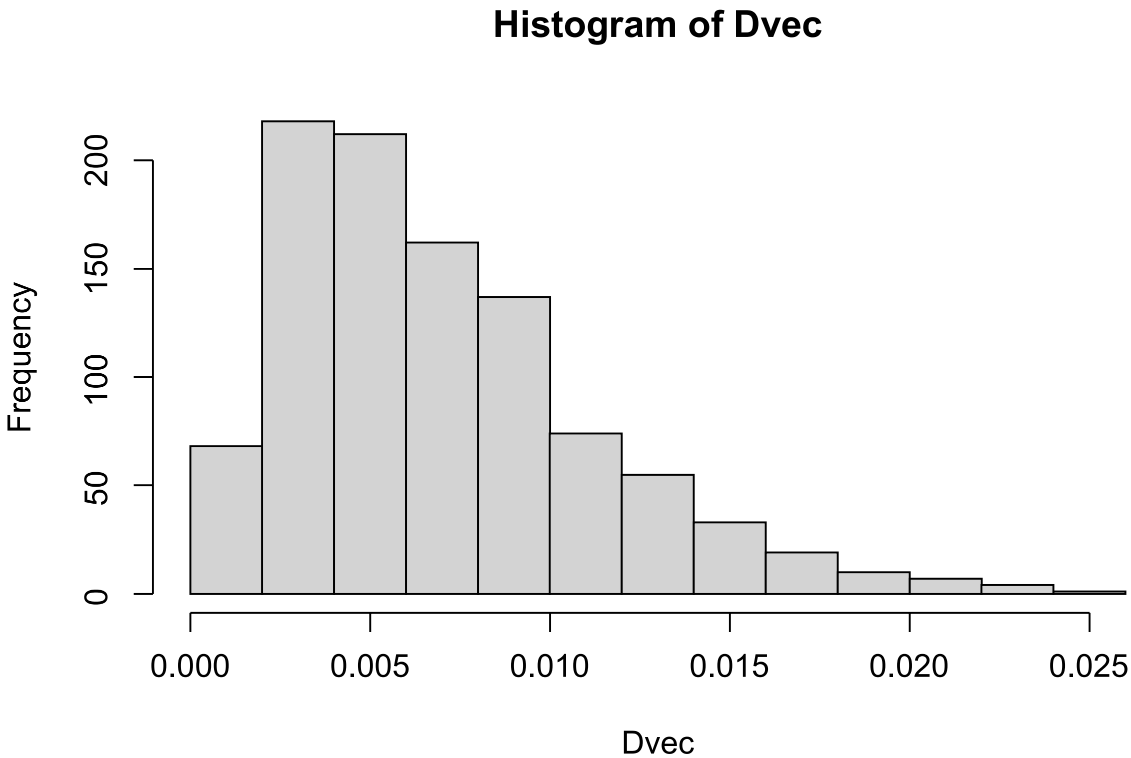

```
## [1] "Method of Moments:"
## [1] "Population Parameter:                lambda = 5"
## [1] "Estimated parameter:lambda_hat = 4.972" ##
## [1] "MLE: "##
[1] 4.972 ##
## [1] "Goodness of fit p-value: " ## [1] 0.606
##
## [1] "-------End of                Poisson ---------------"
##
##   [1]  "------- Exponential -----------------------"
##   [1]  "Paramters of posterior gamma distribution: "
##   [1]  1001.0000      204.8204
##   [1]  "Bayes estimate =          4.88720946991606"
```



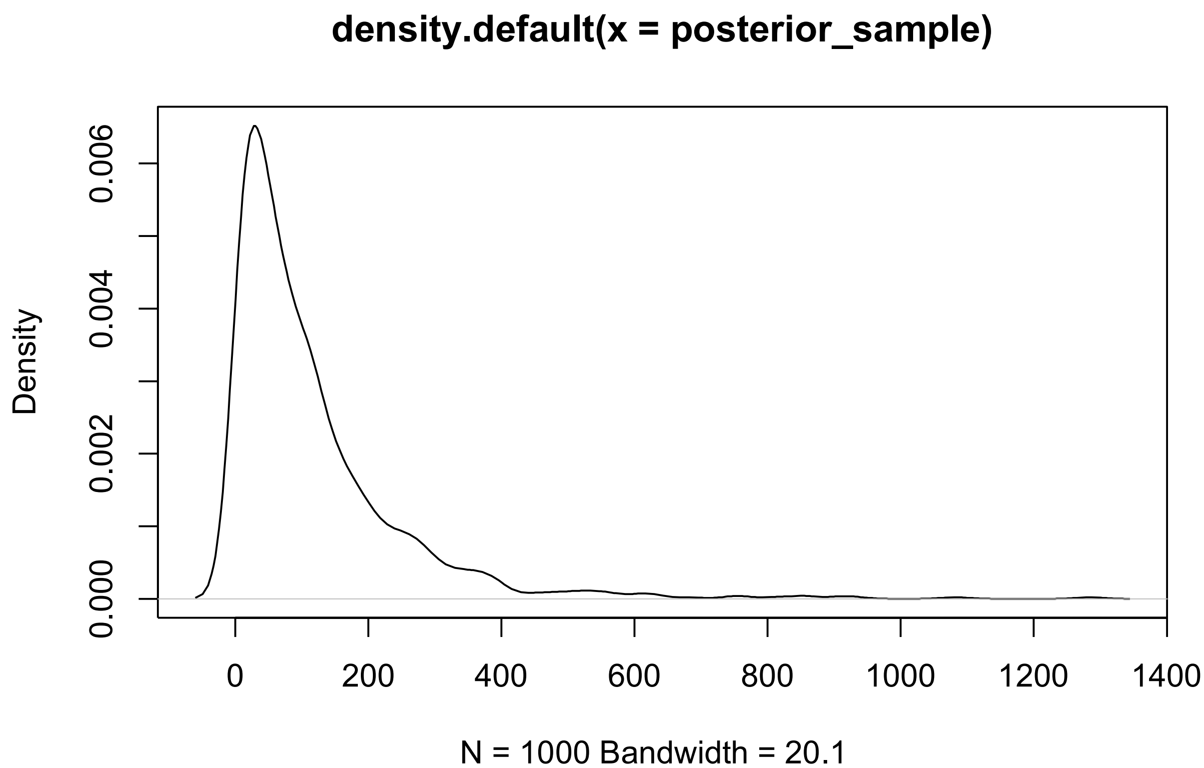

```
## [1] "Method of Moments:"
## [1] "Population Parameter:              beta = 5"
## [1] "Estimated parameter:beta_hat = 4.906" ##
## [1] "MLE: " ##
[1] 4.906281 ##
## [1] "Goodness of fit p-value: " ## [1] 0.065
##
## [1] "-------End of              Exponential ----------------"
##
## [1] "------- Geometric--------------------- "
## [1] "Paramters of posterior beta distribution: "
## [1] 1001 427
## [1] "Bayes estimate =        0.700980392156863"
```



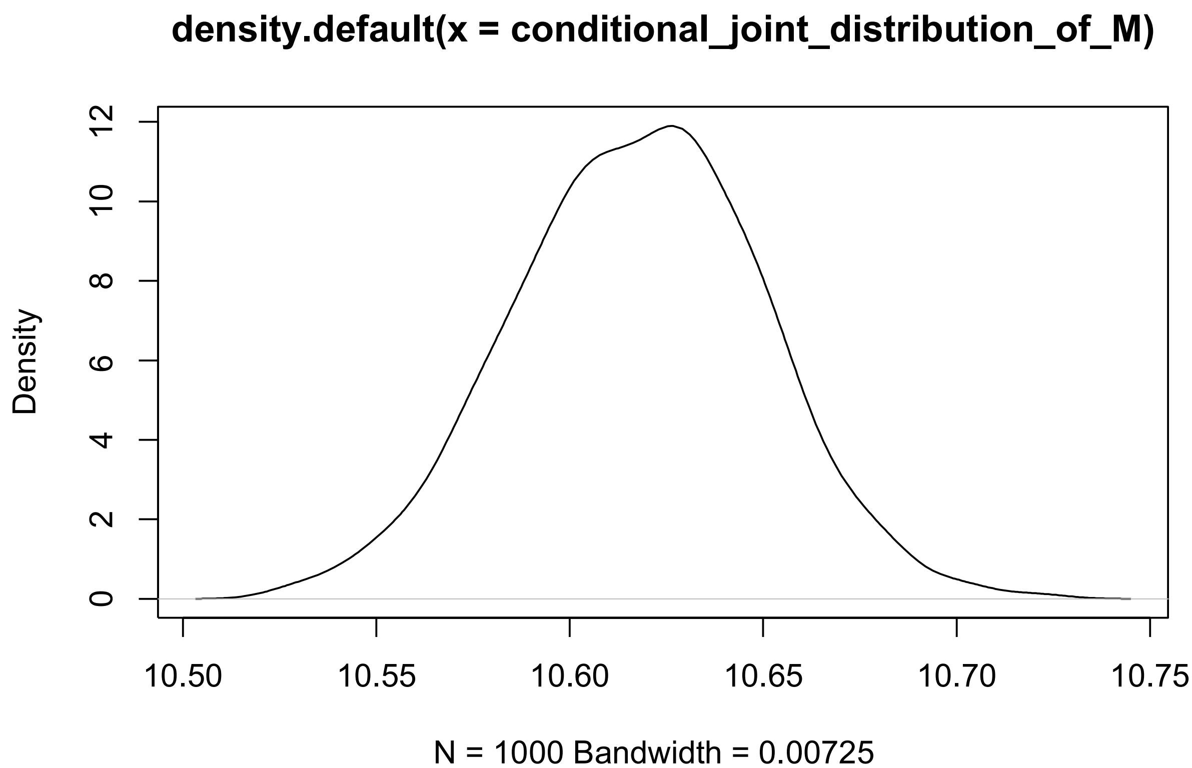
[1] "Method of Moments:"
[1] "Population Parameter: p = 0.7"
[1] "Estimated parameter:p_hat = 1.347" ##
[1] "MLE: " ## [1]
0.7012623

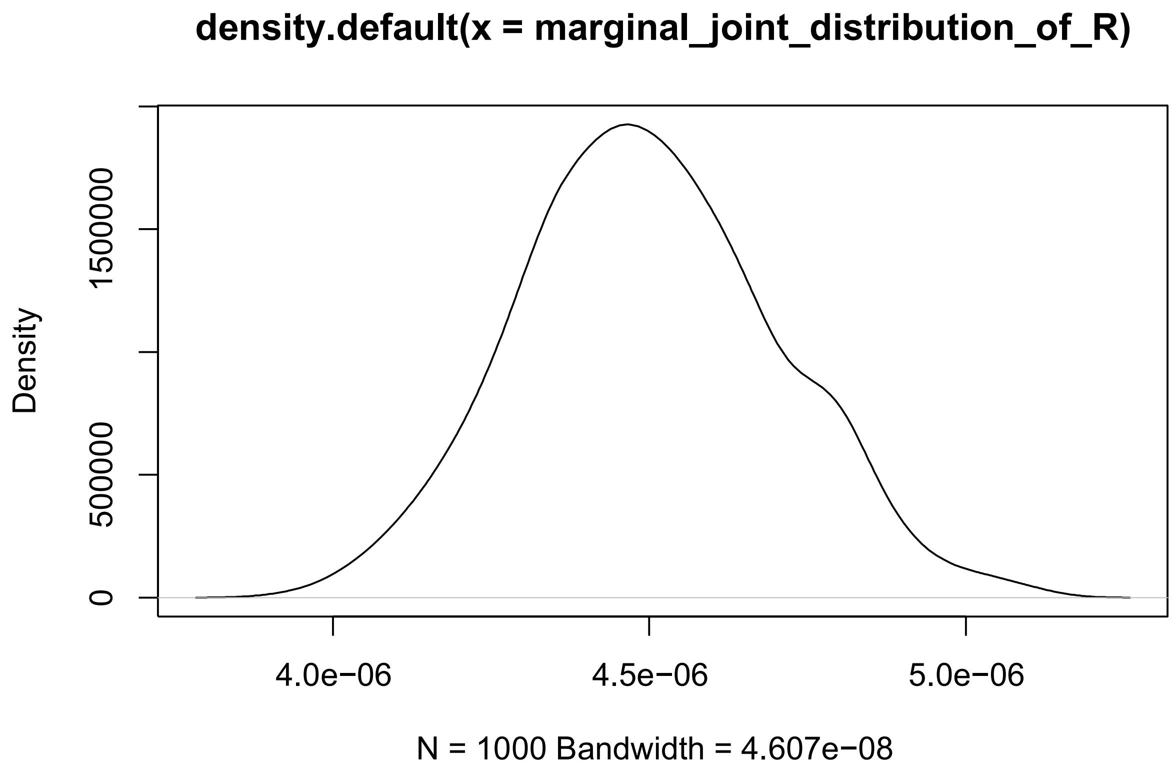



```
## [1] "Goodness of fit p-value: " ## [1] 0.714
##
## [1] "-------End of                Geometric -------------- "
##
## [1] "------- Uniform---------------------"
## [1] "Paramters of posterior pareto distribution: "
## [1]       9.998764 1001.000000
```




```
## [1] "Method of Moments:"
## [1] "Population Parameters:       a = 0            b = 10"
## [1] "Estimated parameters: a_hat = -8.997          b_hat = 19.13" ##

## [1] "MLE: "
## [1] 0.0197762 9.9987642
##
## [1] "Goodness of fit p-value: "
## [1] 0.734
##
## [1] "-------End of      Uniform --------------"
##
## [1] "------- Normal --------------------"
## [1] "The parameters of the conditional posterior normal distribution of M when R=r is:"
## [1]   10.61639 1005.00000
## [1] "The parameters of the marginal posterior gamma distribution of R is:"
## [1]        501 111294200
```


density.default(x = conditional_joint_distribution_of_M)
Density
0
2
4
6
8
10
12
10.50
10.55
10.60
10.65
10.70
10.75
N = 1000 Bandwidth = 0.00725




```
## [1] "Method of Moments:"
## [1] "Population Parameters:        mu = 4              sd = 20"
## [1] "Estimated parameters: mu_hat = 10.649          sd_hat = 20.146"
##
## [1] "MLE: "
## [1] 10.64947 20.14625
##
## [1] "Goodness of fit p-value: "
## [1] 0.098
##
## [1] "-------End of          Normal ------------- "
```

Results

BAYES

Distribution	Parameter 1	Parameter 2	Bayes estimate
Binomial(n, p) (beta)	393	609	0.392215568862275
Poisson(λ) (gamma)	4973	1001	4.96803196803197
Exponential(β) (gamma)	1001	204.8204	4.88720946991606
Geometric(p) (beta)	1001	427	0.700980392156863
Uniform(a, b) (pareto)	9.998764	1001	•
Normal(μ, σ^2) (conditional posterior normal)	10.61639	111294200	•

MOM

Distribution	Actual Parameter 1	Actual Parameter 2	Estimated Parameter 1	Estimated Parameter 2
Binomial(n, p)	n = 1000	n = 0.4	$\hat{n}$ = 1	$\hat{p}$ = 0.392
Poisson(λ)	λ = 5	•	$\hat{\lambda}$ = 4.972	•
Exponential(β)	β = 5	•	$\hat{\beta}$ = 4.906	•
Geometric(p)	p = 0.7	•	$\hat{p}$ = 1.347	•
Uniform(a, b)	a = 0	b = 10	$\hat{a}$ = -8.997	$\hat{b}$ = 19.13
Normal(μ, σ^2)	μ = 4	σ = 20	$\hat{\mu}$ = 10.649	$\hat{\sigma}$ = 20.146

MLE AND GOODNESS OF FIT

Distribution	MLE	Goodness of Fit p-value
Binomial(n, p)	0.392	0
Poisson(λ)	4.972	0.606
Exponential(β)	4.906281	0.065
Geometric(p)	0.7012623	0.714
Uniform(a, b)	0.0197762 | 9.9987642	0.734
Normal(μ, σ^2)	10.64947 | 20.14625	0.098