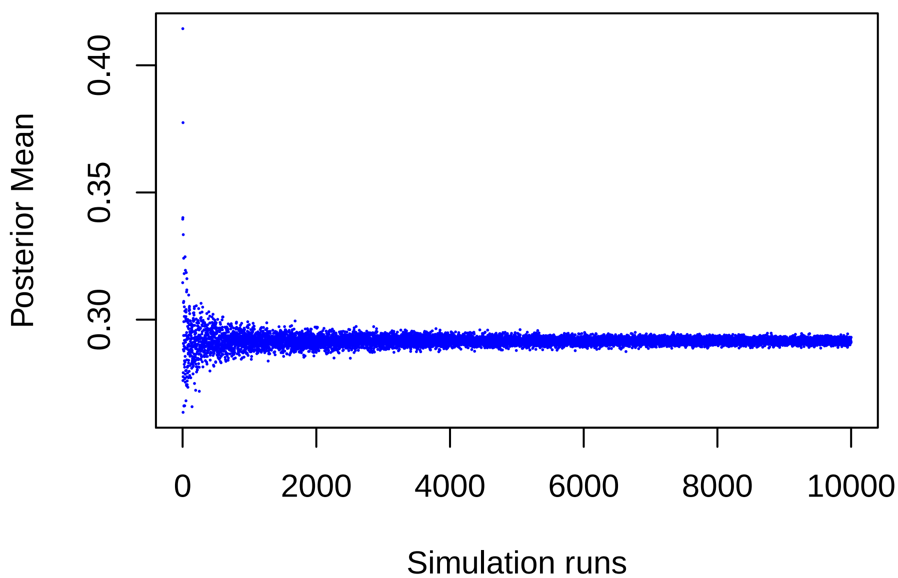
732A91 Bayesian Learning- Computer Lab 1

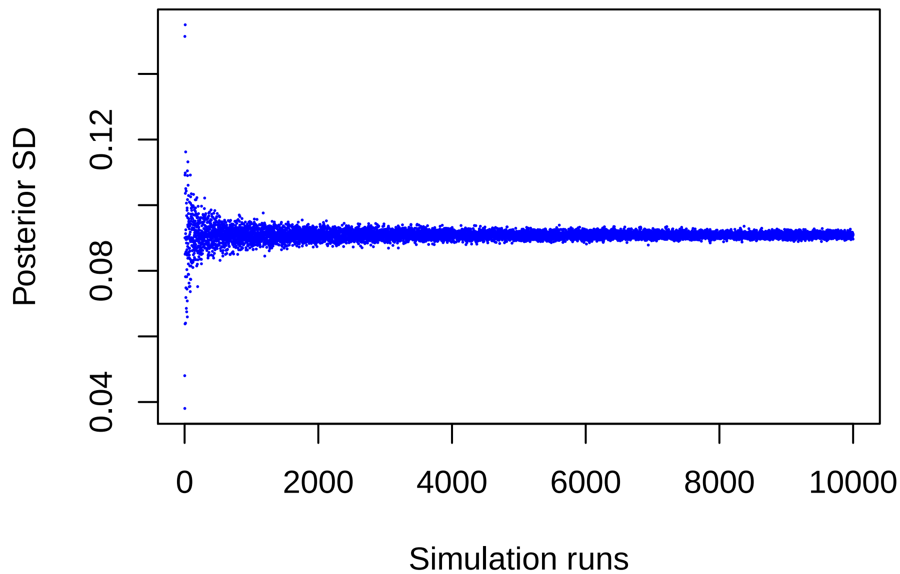
Namita Sharma, Aman Kumar Nayak

4/11/2020

1. Bernoulli

(a) Posterior distribution of theta






```
## Analytical mean of posterior of theta =  0.2916667
## Analytical SD of posterior of theta =  0.09090593
```

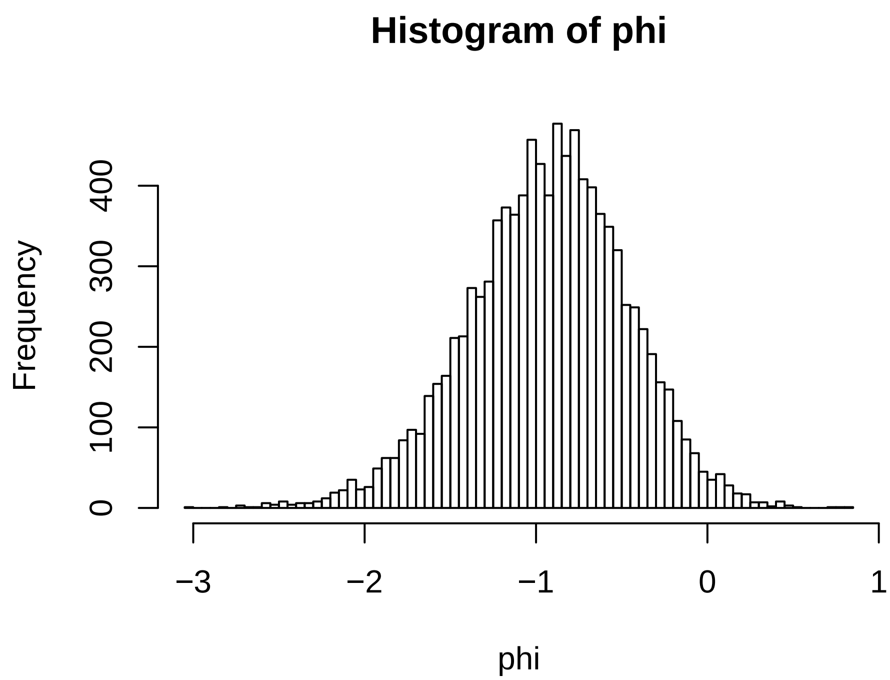
We can see from the graphs that the mean and SD values of the samples drawn from the posterior distribution converge to mean=0.29 and SD=0.09 as the sample size n grow. These values are very close to the true values of mean=0.2916667 and SD=0.09090593. The convergence is achieved quite quickly when sample size n is in a few hundreds but the convergence becomes much more sharper as n grows bigger with fewer fluctutations in the posterior mean and SD values.

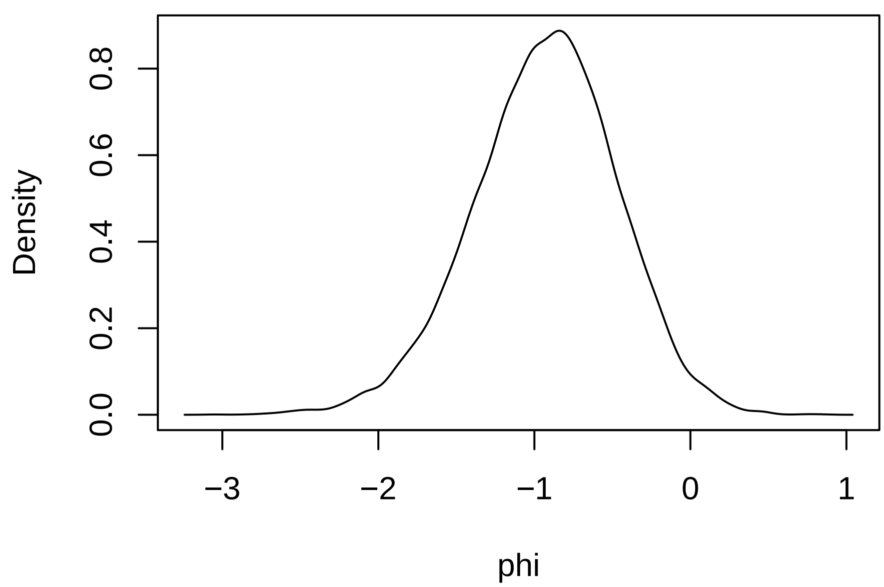
(b) Posterior probability

```
## Pr(theta > 0:3|y) using Simulation =  0.4383
## Pr(theta > 0:3|y) using true distribution =  0.4399472
```

It can be said that the posterior probability $Pr(\theta > 0:3|y)$ computed using simulations is quite close to the true probability.

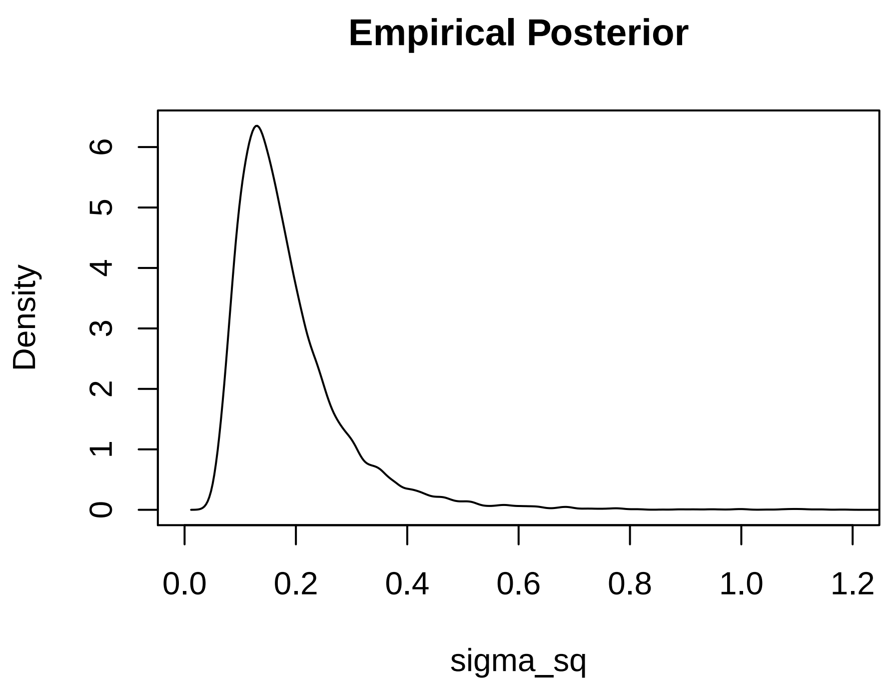
(c) Posterior distribution of log-odds

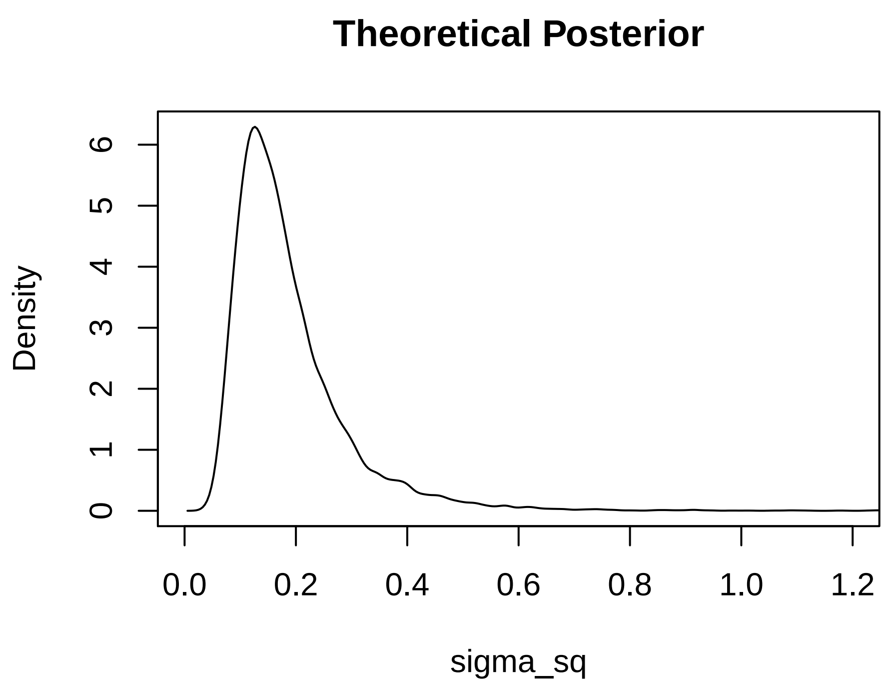




2. Log-normal distribution and the Gini coefficient

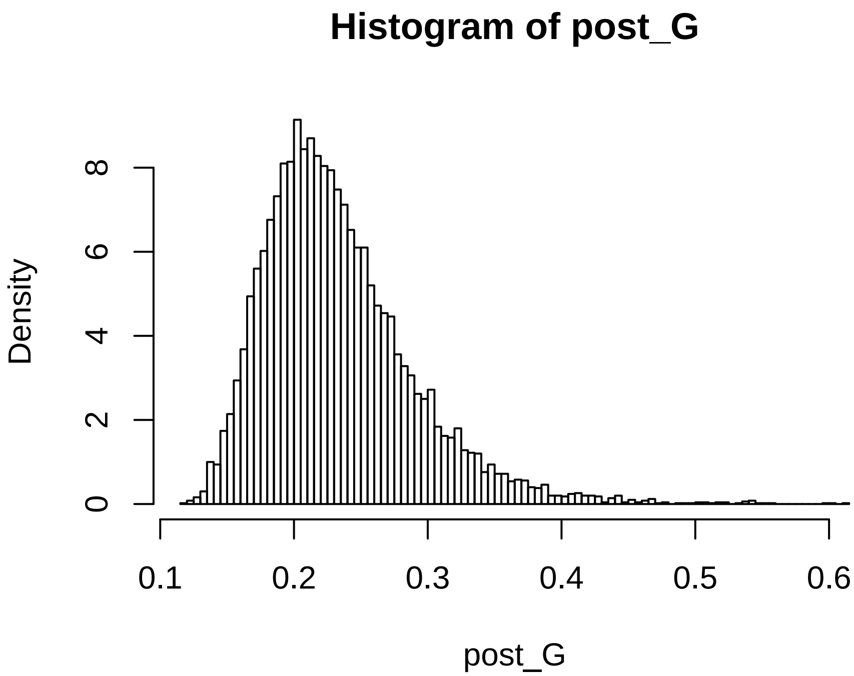
(a) Simulate from posterior of Sigma square

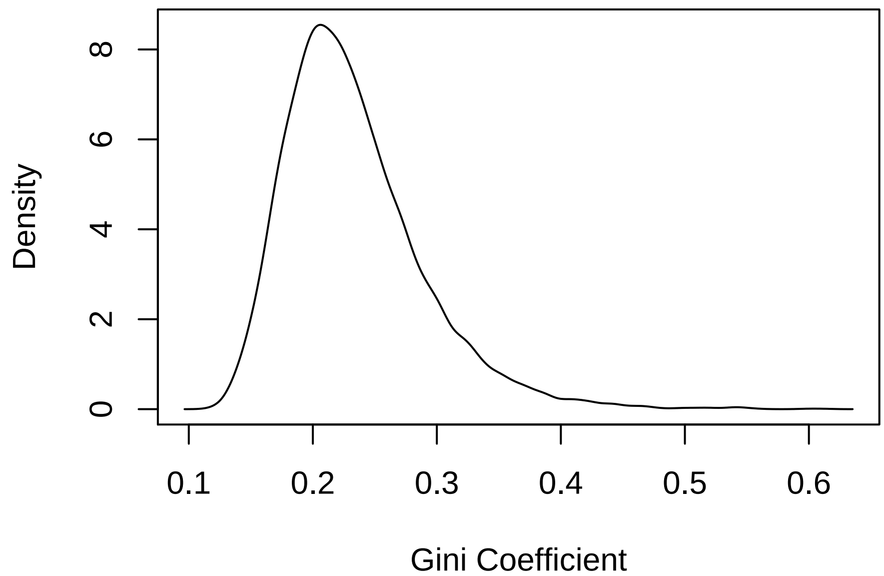




As can be seen from the graphs, the empirical posterior distribution of σ^2 closely resembles the theoretical $Inv - \chi^2(n, \tau^2)$ posterior distribution. The peak and the width of the empirical density distribution matches that of the theoretical distribution.

(b) Gini Coefficient



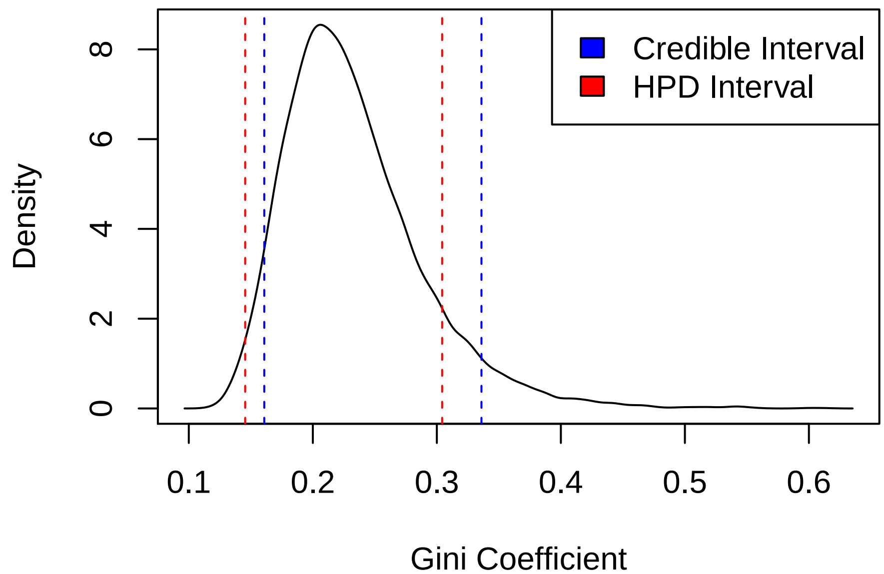


From the posterior of G, we can see that G-value with the highest probability density is approximately 0.2. As it is more close to 0 than to 1, we can conclude that the income distribution is not completely equal and that there is some small inequality among them.

(c) 90% Credible Interval and Highest Posterior Density for G

```
## 90% Equal Tail Interval = [ 0.1608454 : 0.3359687 ]
```

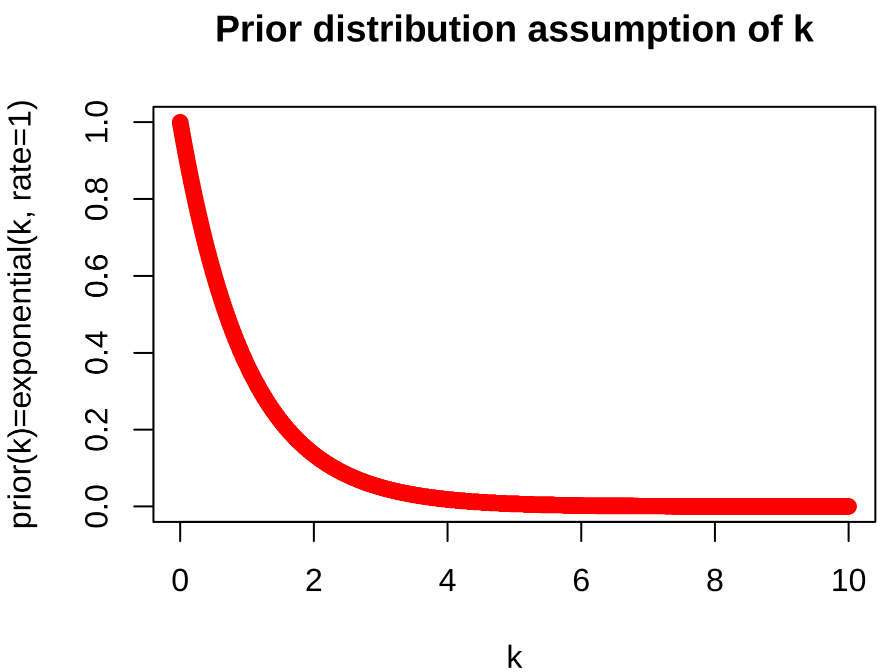
```
## 90% Highest Posterior Density Interval = [ 0.1454683 : 0.304285 ]
```

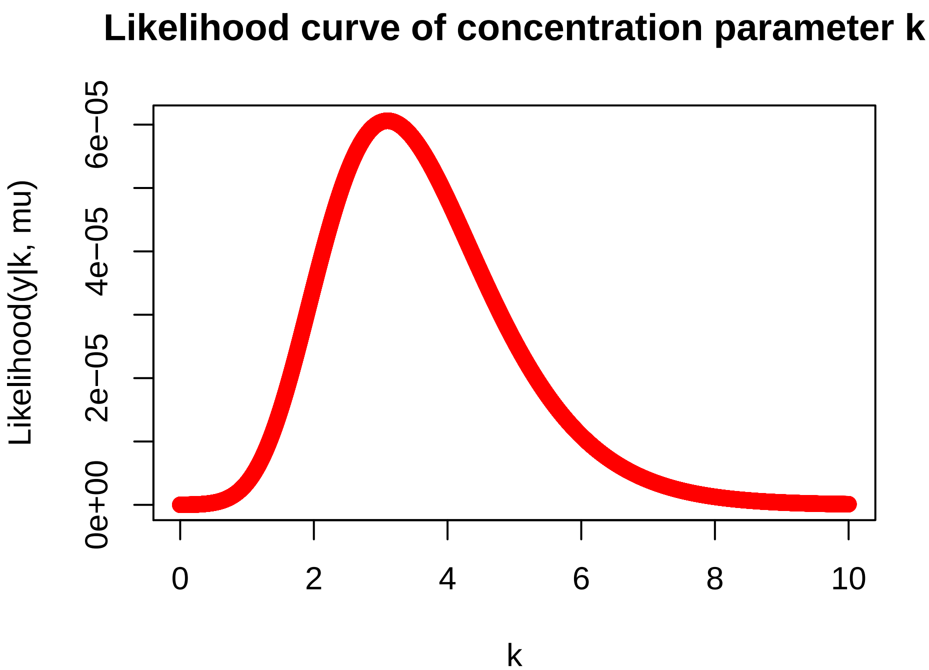


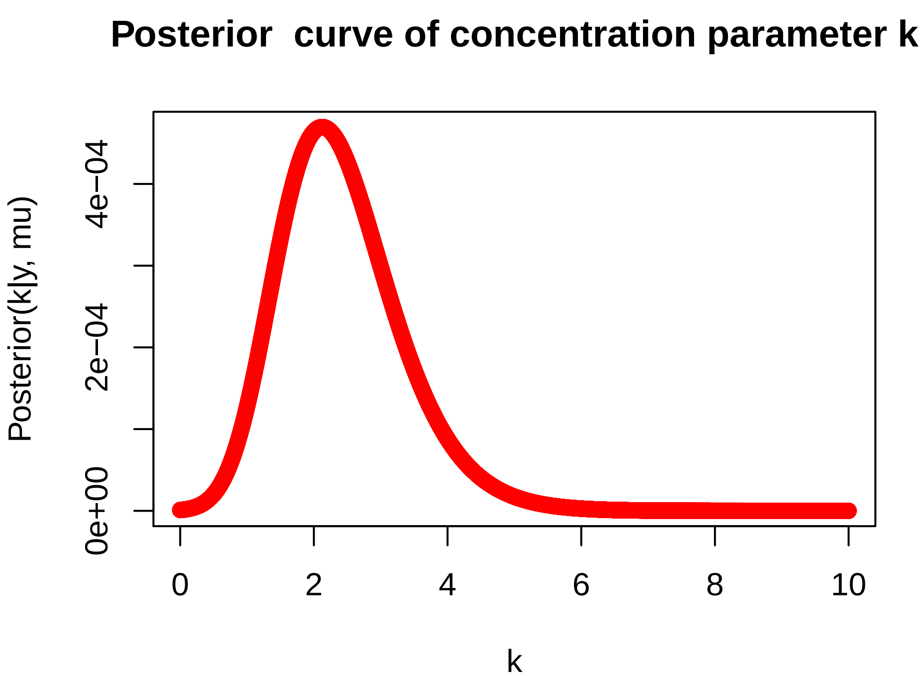
The equal tail interval of the posterior of G is approximately [0.16 : 0.34] which contains 90% of the pdf in the center of the distrbution. Whereas the highest posterior density interval (Remko Duursma 2017) is approximately [0.15 : 0.31] which is slightly to the left of the equal tail interval because it captures the highest 90% of the pdf.

3. Bayesian inference for the concentration parameter in the von Mises distribution

(a) Posterior distribution of k for the wind direction data over a fine grid of k values







The posterior was calculated using the bayes theorem:

$$Posterior(k|y, \mu) = Likelihood(y|k, \mu) * prior(k)$$

The posterior almost looks like a gamma distribution here.

(b) Posterior mode of k from its posterior distribution in (a)

```
## The approximate posterior mode of k =  2.125213
## The posterior density of mode of k =  0.0004695408
```

Appendix

```
##########################################################################
# 1. Bernoulli
##########################################################################

# a) Posterior distribution of theta
# Sample data
n      <- 20
s      <- 5
f      <- n-s

# Prior distribution parameters
alpha0 <- 2
beta0  <- 2

# Simulation from posterior of theta
N    <- 10000
mean <- numeric()
sd   <- numeric()
for (nDraws in 1:N) {
  theta        <- rbeta(nDraws, alpha0+s, beta0+f)
  mean[nDraws] <- mean(theta)
  sd[nDraws]   <- sd(theta)
}

# Plot mean and SD of simulations of theta with increasing number of draws
plot(mean, ylab="Posterior Mean", xlab="Simulation runs", col="blue", pch=16, cex=0.2)
plot(sd, ylab="Posterior SD", xlab="Simulation runs", col="blue", pch=16, cex=0.2)

# Analytical Mean and SD
alpha_new <- alpha0+s
beta_new  <- beta0+f
true_mean <- alpha_new / (alpha_new+beta_new)
true_SD   <- sqrt(alpha_new*beta_new / ((alpha_new+beta_new)^2 * (alpha_new+beta_new+1)))

cat("Analytical mean of posterior of theta = ", true_mean,
    "\nAnalytical SD of posterior of theta = ", true_SD)

# b) Posterior probability
theta     <- rbeta(N, alpha_new, beta_new)
prob      <- sum(theta>0.3) / length(theta)
true_prob <- pbeta(q=0.3, alpha_new, beta_new, lower.tail=FALSE)

cat("Pr(theta > 0:3|y) using Simulation = ",
    prob,
```

```r
    "\nPr(theta > 0:3|y) using true distribution = ",
    true_prob)

# c) Posterior distribution of log-odds
theta <- rbeta(N, alpha_new, beta_new)
phi   <- log(theta / (1-theta))

# Distribution of phi posterior using hist() function
hist(phi, breaks=100)

# Density plot of phi posterior using density() function
phi_pdf <- density(phi)
plot(phi_pdf$x, phi_pdf$y, xlab="phi", ylab="Density", type="l")

########################################################################
# 2. Log-normal distribution and the Gini coefficient
########################################################################
library("geoR")

# a) Simulate from posterior of Sigma square
obs    <- c(44, 25, 45, 52, 30, 63, 19, 50, 34, 67)
n      <- length(obs)
mu     <- 3.7
tau_sq <- sum((log(obs)-mu)^2)/n

# LogNormal posterior from non-Informative prior
logNormal_nonInfoPrior <- function(nDraws, n, tau_sq) {
  # Draw from chisq(n) since we are not losing any df in calculating the mean,
  # i.e. we are given the mean
  X      <- rchisq(nDraws, n)

  # Draw from posterior of sig_sq ~ Inv-chisq(n, tau_sq)
  sig_sq <- n*tau_sq /X

  return(sig_sq)
}

# Empirical posterior distribution of sig_sq
post_sigsq <- logNormal_nonInfoPrior(nDraws=10000, n=n, tau_sq=tau_sq)
post_pdf   <- density(post_sigsq)
plot(post_pdf$x, post_pdf$y, xlab="sigma_sq", ylab="Density", type="l",
     xlim=c(0, 1.2), main="Empirical Posterior")

# Theoretical posterior distribution of sig_sq
sigsq_true   <- geoR::rinvchisq(10000, df=n, scale=tau_sq)
post_truepdf <- density(sigsq_true)
plot(post_truepdf$x, post_truepdf$y, xlab="sigma_sq", ylab="Density", type="l",
     xlim=c(0, 1.2), main="Theoretical Posterior")

# b) Gini Coefficient
gini_coeff <- function(sig_sq) {
  G <- 2 * pnorm(sqrt(sig_sq/2), 0, 1) - 1
  return(G)
```

```r
}
post_G <- gini_coeff(post_sigsq)
hist(post_G, breaks=100, probability=TRUE)

# Density plot of G posterior using density() function
G_pdf <- density(post_G)
plot(G_pdf$x, G_pdf$y, type="l", xlab="Gini Coefficient", ylab="Density")

# c) 90% Credible Interval and Highest Posterior Density for G
credInterval <- function(credible=0.9) {
  eq_tail <- (1-credible)/2                                  # tail region
  tail    <- eq_tail*length(post_G)                          # tail % of the posterior samples
  CredI   <- post_G[order(post_G)][tail:(length(post_G)-tail)] # credible interval
  CredI_L <- CredI[1]                                        # lower limit
  CredI_U <- CredI[length(CredI)]                            # upper limit

  return(c(CredI_L, CredI_U))
}

CredI <- credInterval(credible=0.9)
cat("90% Equal Tail Interval = [", CredI[1], ":", CredI[2], "]")

# c) Highest Posterior Density Interval for post_G
HPD <- function(emp_pdf, prob=0.9) {
  # Mathematical integration of the empirical density curve can be approximated using Riemann sum
  x      <- emp_pdf$x                  # 512 points where the density is estimated
  y      <- emp_pdf$y                  # 512 density values estimated for x
  dx     <- emp_pdf$x[2]-emp_pdf$x[1]  # Spacing or bin size
  C      <- sum(emp_pdf$y) * dx        # Normalizing constant C = Riemann sum approx. of area under th
  mode   <- x[which.max(y)]            # Mode of the density curve
  domain <- range(x)                   # Domain of the posterior pdf

  # Area under the curve to the right of x=a and left of x=b or Pr(a<x>b)
  prob_ab <- function(a, b) {
    p.unscaled <- sum(y[x > a & x < b]) * dx
    p.scaled   <- p.unscaled / C

    return(p.scaled)
  }

  # Given a and prob, invert to find b
  invert_prob_ab <- function(a, prob) {

    # Cost function to find the interval [a, b] that contains prob probability
    optim_b <- function(b, a) {
      p.area <- prob_ab(a, b)
      return((prob - p.area)^2 )
    }
    b <- optimize(optim_b, c(a, domain[2]), a=a)$minimum # Search over pdf curve to the right of a
    return(b)
  }

  # Given prob, find the shortest [a ,b] that contains prob
```

```r
  shortest_ab <- function(prob) {

    # Cost function to find the shortest interval [a, b] that contains prob probability
    optim_a <- function(a) {
      b <- invert_prob_ab(a, prob=prob)
      return (b-a)
    }
    a <- optimize(optim_a, c(domain[1], mode))$minimum  # Search over pdf curve to the left of mode
    b <- invert_prob_ab(a, prob=prob)

    return(c(a, b))
  }

  HPD_Int  <- shortest_ab(prob=prob)              # Highest posterior density interval
  HPD_prob <- prob_ab(a=HPD_Int[1], b=HPD_Int[2]) # Verify Interval contains 90% of the highest posteri
  return(HPD_Int)
}

HPD_Int  <- HPD(emp_pdf=G_pdf, prob=0.9)
cat("90% Highest Posterior Density Interval = [", HPD_Int[1], ":", HPD_Int[2], "]")

# Graphical representation of CredI and HPD on G distribution curve
plot(G_pdf$x, G_pdf$y, type="l", xlab="Gini Coefficient", ylab="Density")
abline(v=CredI, col="blue", lty="dashed")
abline(v=HPD_Int, col="red", lty="dashed")
legend("topright", c("Credible Interval", "HPD Interval"), fill=c("blue", "red"))

########################################################################
# 3. Bayesian inference for the concentration parameter in the von
#    Mises distribution
########################################################################
y <- c(-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02)

# a) Posterior distribution of k for the wind direction data over a
#    fine grid of k values
prior <- function(k, rate=1) {
  return(dexp(k, rate))
}

likelihood <- function(y, mu=2.39, k) {
  n <- length(y)
  l <- exp(k*sum(cos(y-mu))) / (2*pi*besselI(k, nu=0))^n
  return(l)
}

k_grid      <- seq(0, 10, length.out=10000)
k_posterior <- likelihood(y, k=k_grid) * prior(k=k_grid, rate=1)
k_posterior <- k_posterior / sum(k_posterior)

# Assumed prior distibution of k
plot(k_grid, prior(k=k_grid, rate=1), main="Prior distribution assumption of k",
     xlab="k", ylab="prior(k)=exponential(k, rate=1)", col="red")
```

```r
# Likelihood curve tells us which is the MLE estimate of k
plot(k_grid, likelihood(y, k=k_grid), main="Likelihood curve of concentration parameter k",
     xlab="k", ylab="Likelihood(y|k, mu)", col="red")

# Posterior distribution of k gives us the uncertainity
plot(k_grid, k_posterior, main="Posterior  curve of concentration parameter k",
     xlab="k", ylab="Posterior(k|y, mu)", col="red")

# b) Posterior mode of k from its posterior distribution in (a)
mode       <- which.max(k_posterior)
k_mode     <- k_grid[mode]
k_mode_pdf <- max(k_posterior) # values of posterior pdf at k_mode

cat("The approximate posterior mode of k = ", k_mode,
    "\nThe posterior density of mode of k = ", k_mode_pdf)
```

References

Remko Duursma. 2017. "Stack Overflow." https://stackoverflow.com/questions/45702886/can-we-use-base-r-to-find-the-95-of-the-area-under-a-curve.