

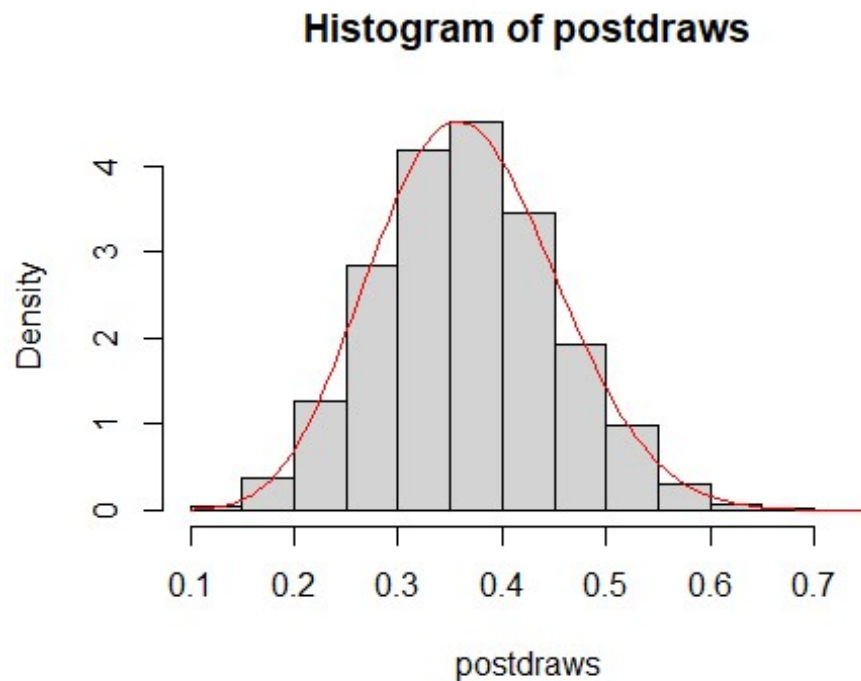
## lab1

Alexander Bois(alebo256) & Daniel Bissessar(danbi675)

### Task 1

a)

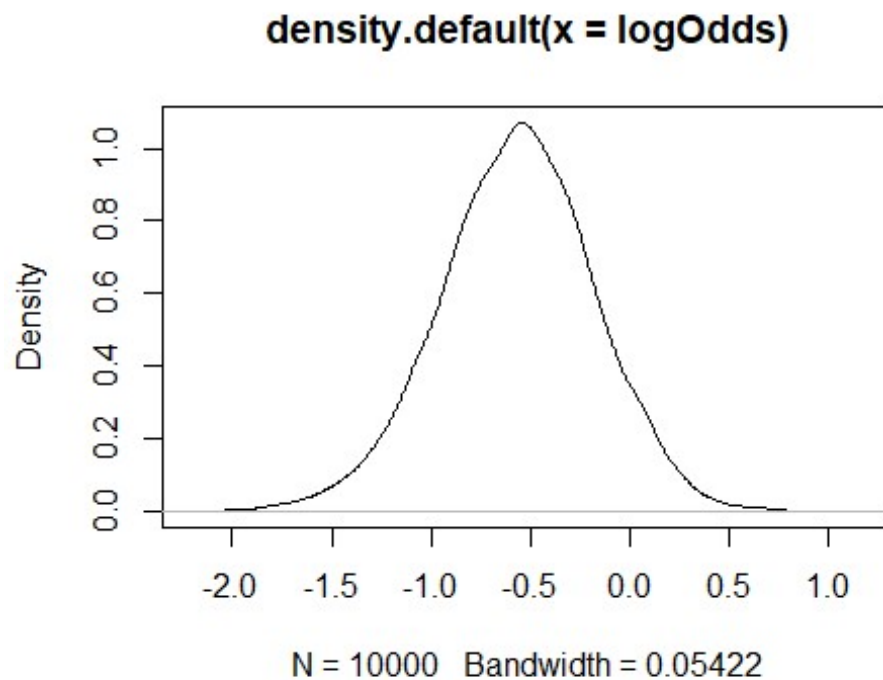
```
set.seed(12345)
draws = 10000
x <- seq(0.001, 0.999, by=0.001)
postdraws <- rbeta(draws, 11,19)
hist(postdraws, freq = FALSE)
curve(dbeta(x,11,19),add=TRUE,col="red")
```



b) As can be seen the exact and simulated values are very close to each other.

```
exactVal <- pbeta(0.4, 11, 19)
simVal <- mean(postdraws<0.4)
print('Exact value:')
## [1] "Exact value:"
exactVal
```

```
## [1] 0.6573346
print('Simulated value:')
## [1] "Simulated value:"
simVal
## [1] 0.6617
c)
logOdds <- log(postdraws/(1-postdraws))
den <- density(logOdds)
plot(den)
```



## Task 2

a)

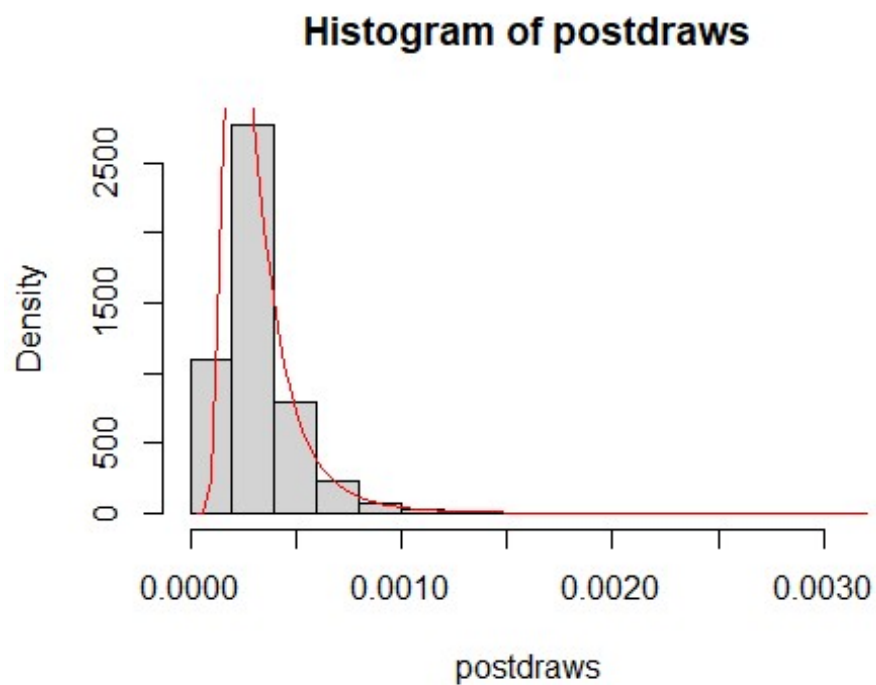
```
draws <- 10000
y <- c(38,20,49,58,31,70,18,56,25,78)

postdraws <- rchisq(draws, 10)
mu <- 3.8
tao <- sum((log(y)-mu)^2)/(draws)
var <- ((draws-1)*tao)/postdraws
```

```

postdraws <- (10*tao)/postdraws
hist(postdraws, freq=FALSE)
dinvchisq <- function(x, n, t) {
  a <- n/2
  b <- n*t/2
  (b^a)/gamma(a) * x^(-a-1) * exp(-b/x)
}
x <- seq(0.00001, 0.1, by=0.001)
curve(dinvchisq(x,10,tao),add=TRUE,col="red")

```

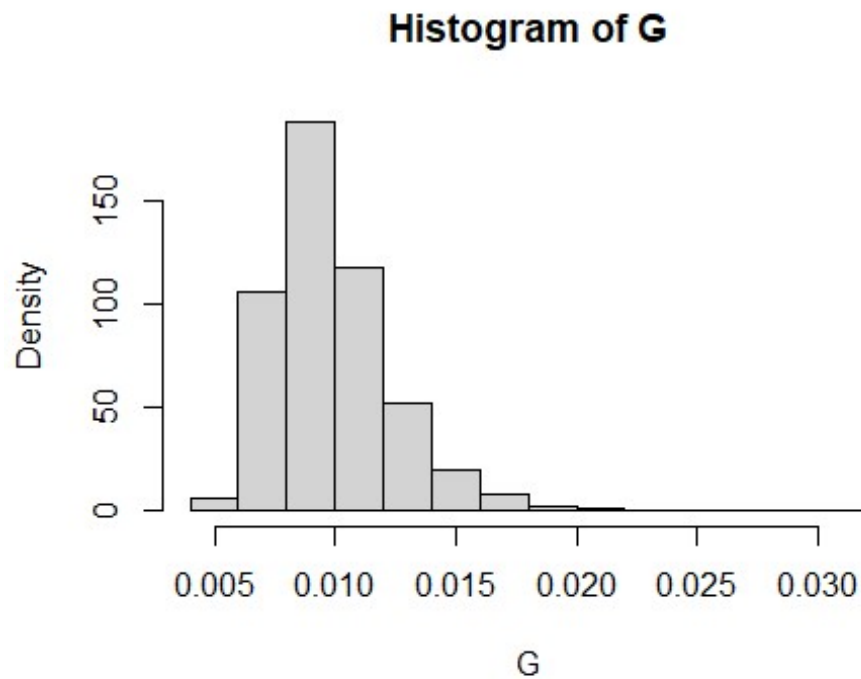


b)

```

G <- -1+2*pnorm(sqrt(postdraws)/sqrt(2))
hist(G, freq = FALSE)

```



c) As can be seen the intervals are very close in size.

```
cred_interval <- quantile(G, probs=c(0.05,0.95))
newG <- G[which(cred_interval[2]>G)]
newG <- newG[which(newG>cred_interval[1])]

den <- density(G)
dn <- cumsum(den$y)/sum(den$y)
li <- which(dn>=0.05)[1]
ui <- which(dn>=0.95)[1]
HPDI <- den$x[c(li,ui)]
print('Equal tail credible interval:')

## [1] "Equal tail credible interval:"

cred_interval

##           5%           95%
## 0.006741373 0.014383635

print('HPDI:')

## [1] "HPDI:"

HPDI

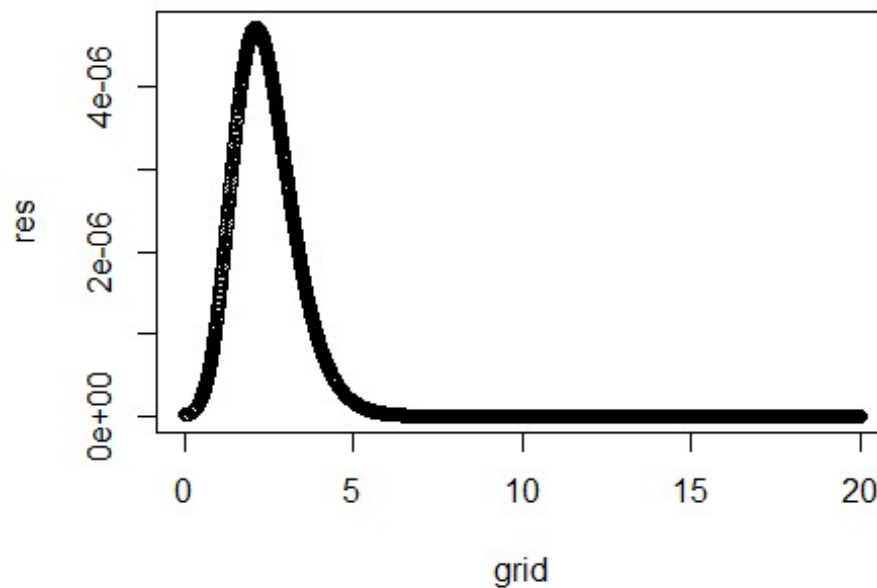
## [1] 0.006663427 0.014457539
```

### Task 3

a)

```
y <- c(-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02)
mu <- 2.39
likelihood <- function(k){
  return(prod(exp(k*cos(y-mu))/(2*pi*besselI(k, 0))))
}

grid <- seq(0,20,0.01)
prior <- dexp(grid)
res <- rep(0, length(grid))
i <- 1
for (val in grid){
  res[i] <- likelihood(val)
  i <- i+1
}
res <- res*prior
plot(grid,res)
```



b) We assume that distribution is symmetric.

```
kappaind <- which.max(res)
posteriorMode <- grid[kappaind]
print('Approximate Posterior Mode:')
## [1] "Approximate Posterior Mode:"
posteriorMode
## [1] 2.12
```