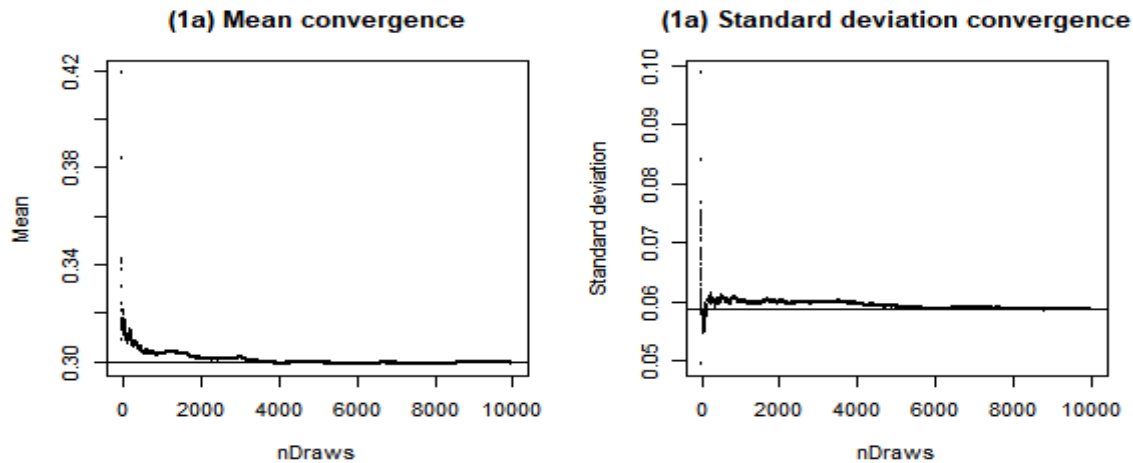


# Bayesian Learning - Computer Lab 1

Ludvig Knast ludkn080 & Erik Jareman erija971

1. a)



```
# --- Assignment 1 ---
alpha = 5
beta = 5
s = 13
trials = 50
nDraws = 10000
newAlpha = (alpha + s)
newBeta = (beta + trials - s)

# 1a
par(mfrow=c(3,3))

trueMean = newAlpha / (newAlpha + newBeta)

rnd = rbeta(nDraws, newAlpha, newBeta)

plot(cumsum(rnd) / seq_along(rnd), cex = 0.3, main = "(1a) Mean convergence",
     xlab = "nDraws", ylab = "Mean")

abline(h = trueMean)

trueSD = sqrt((newAlpha * newBeta) /
              (((newAlpha+newBeta)^2)*(newAlpha+newBeta+1)))

x <- c()
for (index in 1:nDraws)
{
  x <- c(x, sd(rnd[1:index]))
}
plot(x, cex = 0.3, main = "(1a) Standard deviation convergence",
     xlab = "nDraws", ylab = "Standard deviation")
abline(h = trueSD)
```

b)

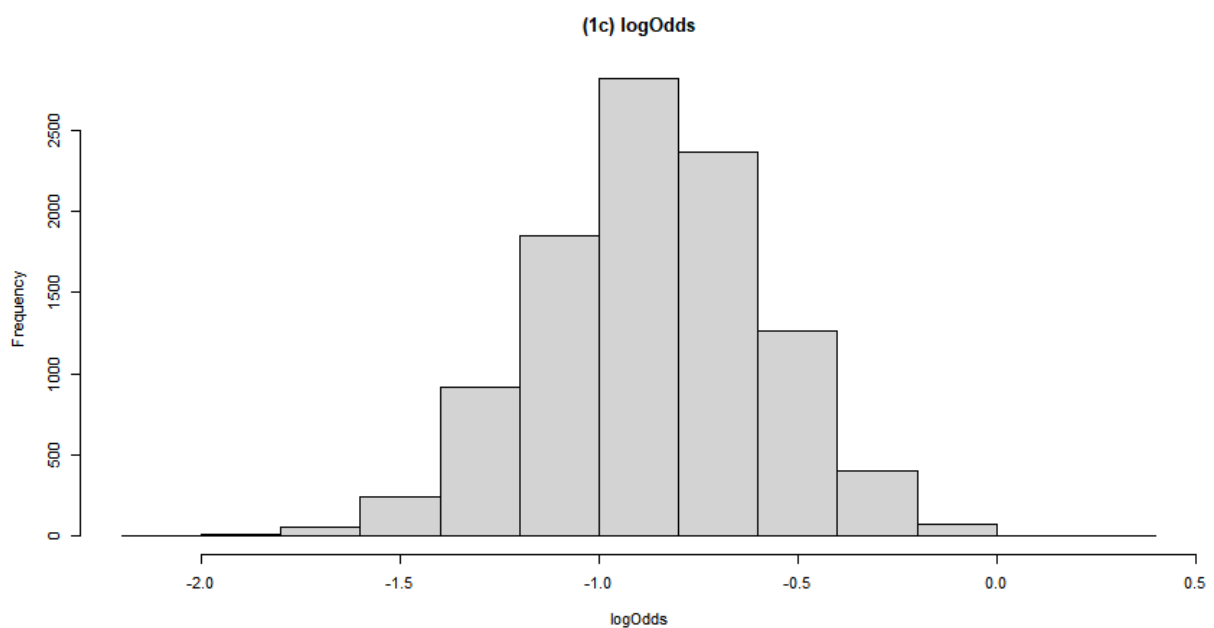
Results (computed probability vs exact):

Postprob 0.5212

Exact prob 0.5150226

```
# 1b
postProb <- sum(rnd < 0.3)/nDraws
exactProb <- pbeta(0.3, newAlpha, newBeta)
cat(" Postprob", postProb, "\n", "Exact prob", exactProb)
```

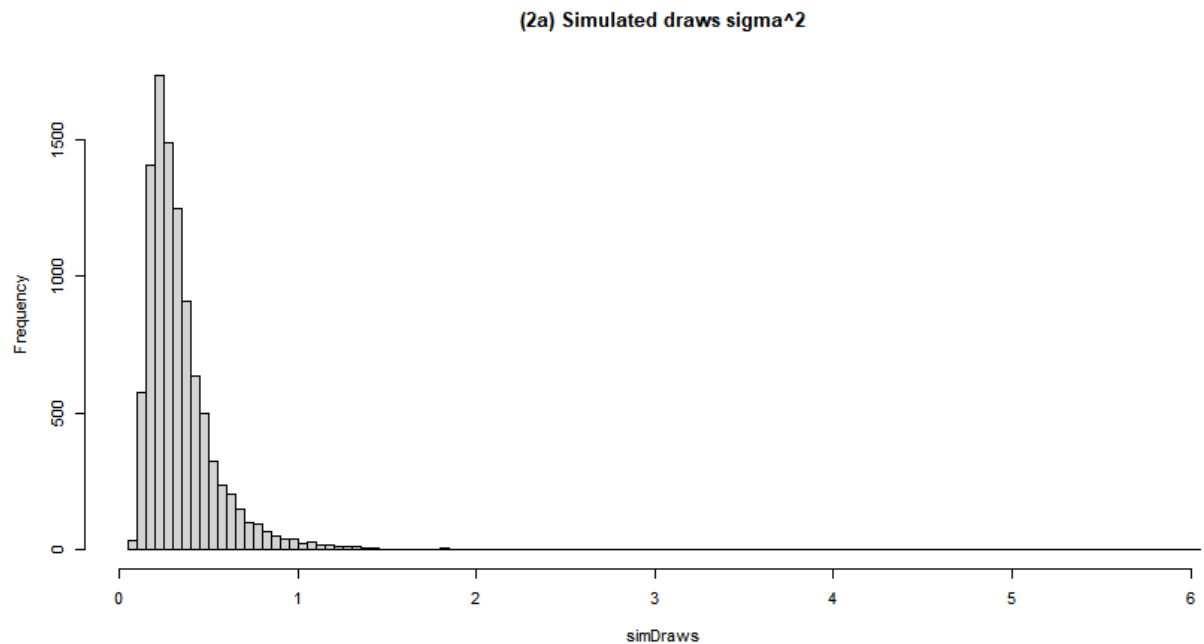
c)



```
# 1c
#library(rcompanion)
logOdds <- log(rnd/(1-rnd))
#plotNormalHistogram(logOdds)
hist(logOdds, main="(1c) logOdds")
```

2.

a)



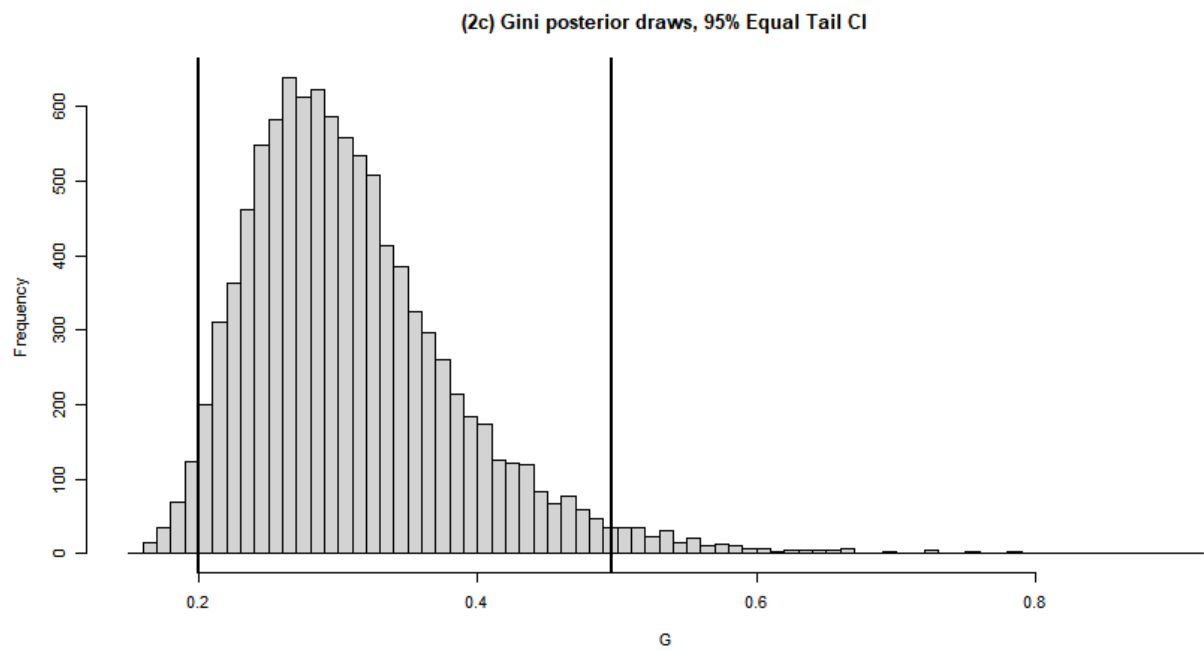
```
# --- Assignment 2 ---
nDraws = 10000
my <- 3.5
obs <- c(33, 24, 48, 32, 55, 74, 23, 76, 17)
nObs = length(obs)

# 2a
rndChi <- rchisq(n=nDraws, df=nObs)
simDraws <- nObs * (sum((log(obs) - my)^2)/nObs) / rndChi # kanske (n-1) ???
hist(simDraws, 100, main="(2a) Simulated draws sigma^2")
```

b)

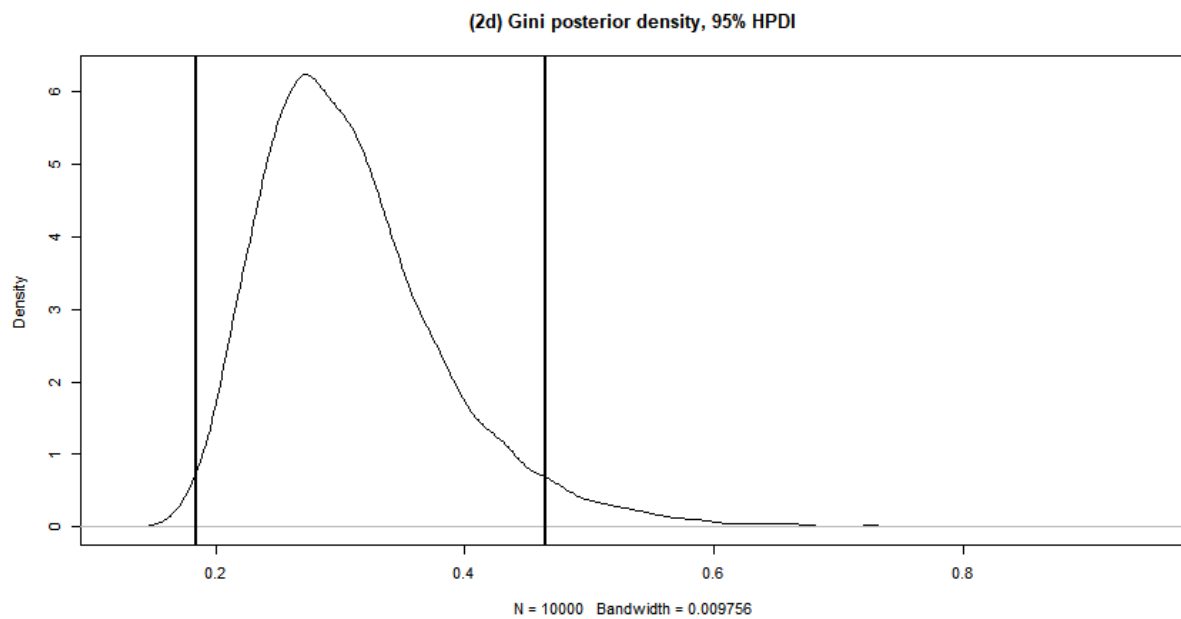
```
# 2b
G <- 2 * pnorm(sqrt(simDraws)/sqrt(2), mean=0, sd=1) - 1
```

c)



```
# 2c
sortG <- sort(G)
G_hi = sortG[nDraws*0.975]
G_lo = sortG[nDraws*0.025]
hist(G, 100, main="(2c) Gini posterior draws, 95% Equal Tail CI")
abline(v=G_lo, lwd=2)
abline(v=G_hi, lwd=2)
```

d)

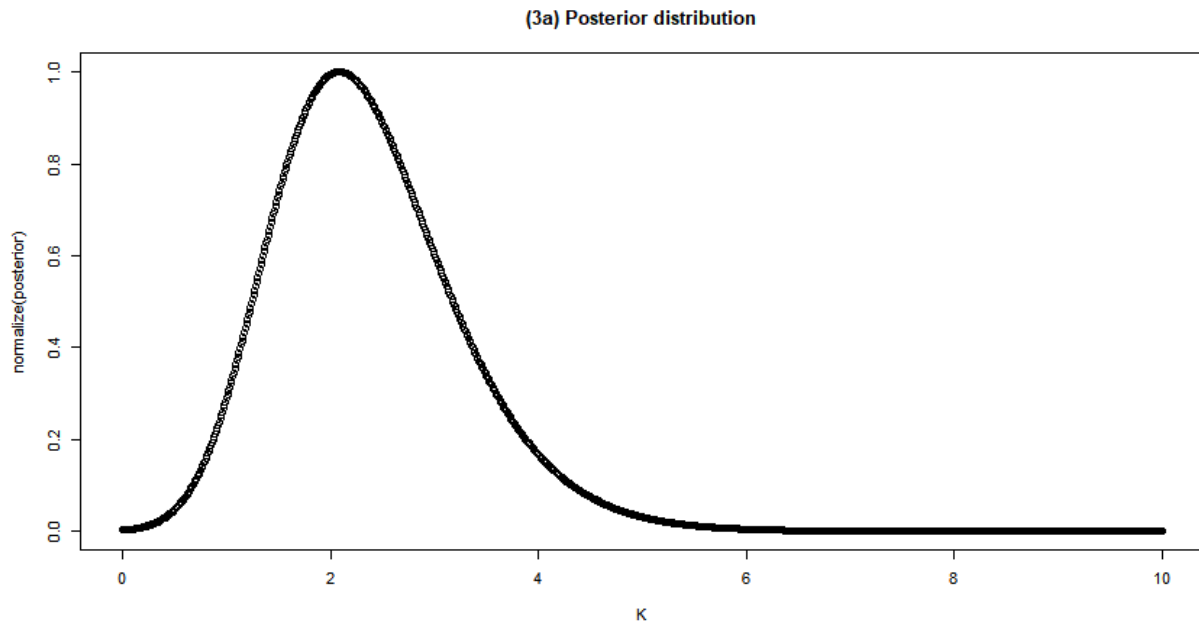


```
# 2d
denseG = density(G)
denseG$x = denseG$x[order(denseG$y, decreasing=FALSE)]
denseG$y = sort(denseG$y, decreasing=FALSE) # sortera båda samtidigt...
tot = sum(denseG$y)
cumSum = 0
i = 1
while ((cumSum / tot) < 0.05) {
  cumSum = cumSum + denseG$y[i]
  i = i + 1
}
interval = denseG$x[i:length(denseG$x)]
G_hi_HPDI = max(interval)
G_lo_HPDI = min(interval)

plot(density(G), main="(2d) Gini posterior density, 95% HPDI")
abline(v=G_hi_HPDI, lwd=2)
abline(v=G_lo_HPDI, lwd=2)
```

3.

a)



```
# --- Assignment 3 ---
obs = c(1.83, 2.02, 2.33, -2.79, 2.07, 2.02, -2.44, 2.14, 2.54, 2.23)
my <- 2.51

# 3a
K = seq(from=0, to=10, by=0.01)
prior = dexp(K, rate=1)

lh = c()
for (index in 1:length(K)) {
  lh[index] = prod(exp(K[index] * cos(obs - my)) / (2 * pi * besseli(K[index],
nu=0)))
}

posterior = lh * prior

normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}
plot(K, normalize(posterior), main="(3a) Posterior distribution")
```

b)

Approximate posterior mode (K): 2.09.

```
#3b
maxPost = which.max(posterior)
cat(K[maxPost])
```