# Bayesian Learning

## Computer Lab 1

# 1. Bernoulli … again

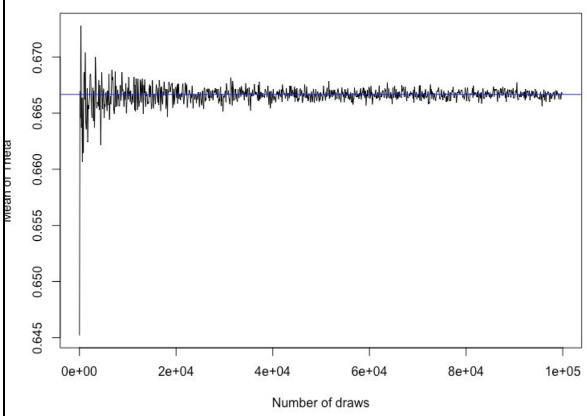
We assume that y1, ..., yn|θ ~ Bern(θ), and that we have obtained a sample with s = 14 successes in n = 20 trials. Assume a Beta( $\alpha_0$ , $\beta_0$ ) prior for θ and let $\alpha_0$ = $\beta_0$ = 2.
f = n - s = 20 - 14 = 6

a)
posterior_alpha: 2 + 14 = 16
posterior_beta: 2 + 6 = 8
True posterior mean = $\frac{16}{16+8} = \frac{16}{24} = 0,666667$
True posterior standard deviation = $\sqrt{\frac{16 * 8}{(16+8)^2 * (16+8+1)}} = 0.09428$

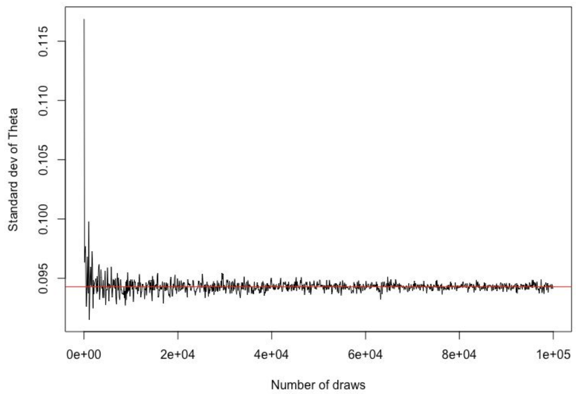
To verify that the mean and standard deviation converges towards the true values as the number of draws grows large, different sized draws (nDraws = 10, 20, … , 100000) from the posterior distribution $\theta \mid y \sim Beta\,(\alpha_0$ + s, $\beta_0$ + f ) was done. The mean and standard deviation was then calculated for each draw and plotted on two separate plots. The plots verify that as the number of draws grows large, the mean and standard deviation converges towards the true values.

```r
theta.means <- c()
theta.sd <- c()
theta.n <- c()
for (n in seq(10, 100000, 100)) {
  thetas <- rbeta(n, posterior_alpha, posterior_beta) # Generate Thetas
  theta.means <- append(theta.means, mean(thetas)) # Calculate mean and add to vector
  theta.sd <- append(theta.sd, sd(thetas)) # Calculate standard deviation and add to vector
  theta.n <- append(theta.n, n) # Add number of draws to vector
}
```

**Graphical representation of mean and standard deviation of theta.**




*Mean of thetas, as the number of draws increases*




*Standard deviation of theta, as the number of draws increases*

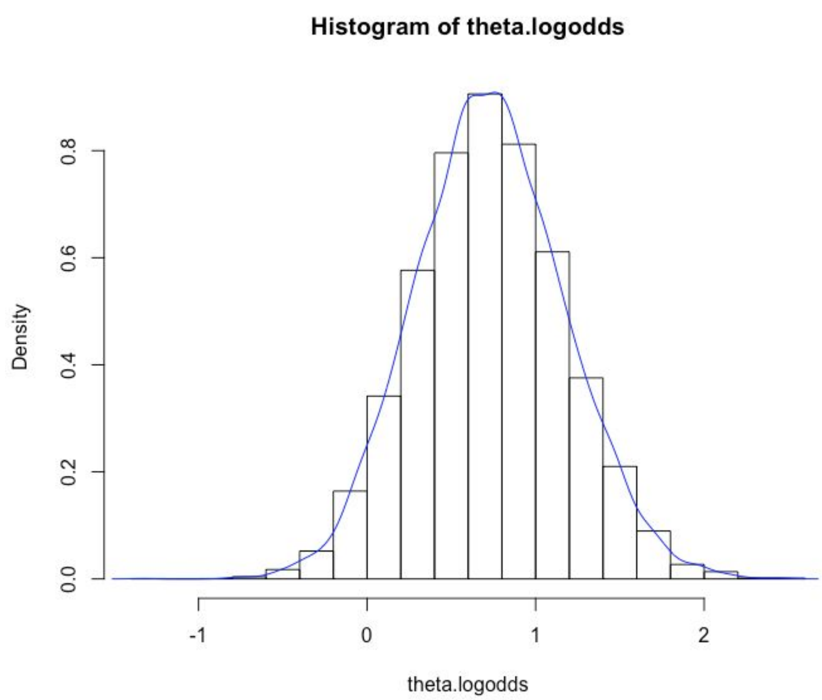
b)
To compute the posterior probability Pr( $\theta < 0.4 \mid y$ ) by simulation, we start of by simulating 10000 draws from the beta-distribution, posterior_alpha = 16, posterior_beta = 8. By dividing the number of simulated theta-values below 0.4 with the total number of simulated theta-values (10000), the posterior probability is obtained.

Simulated posterior probability: 0.0045
Actual posterior probability: 0.003972681

```
theta.draws <- rbeta(10000, posterior_alpha, posterior_beta) # Draw 10000 numbers
theta.no_smaller <- ifelse(theta.draws < 0.4, 1, 0) # If draw < 0.4 -> 1, else -> 0
theta.prob_smaller <- sum(theta.no_smaller)/length(theta.draws) # Calculate probability of drawn number being < 0.4
print(theta.prob_smaller)
print(pbeta(0.4,posterior_alpha, posterior_beta)) # Actual probability of b
```

c)
To compute the posterior probability of the log-odds φ = log( $\frac{\theta}{1-\theta}$ ) we simply use the 10000 draws done in task b and insert them in the log-odds function. The result was then plotted on a histogram. The density function was then plotted on top of the histogram.

```
theta.logodds <- log(theta.draws/(1-theta.draws))
theta.logodds.density <- density(theta.logodds)
hist(theta.logodds, probability = TRUE)
lines(theta.logodds.density, col='blue')
```



# 2. Log-normal distrubtion and the Gini coefficient

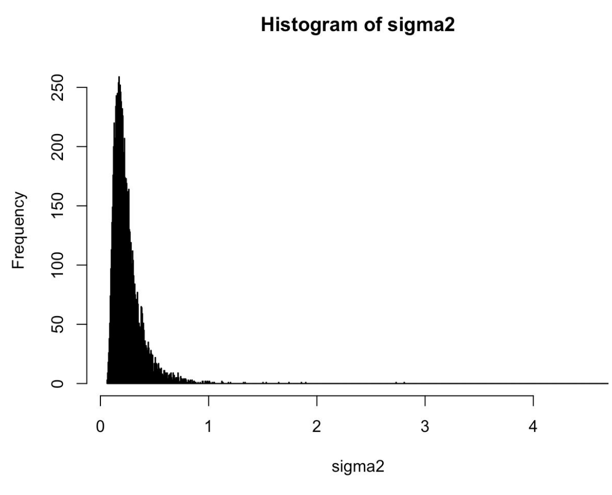
Given observations y of salaries modelled as *logN*(μ, $\sigma^2$) and density function:

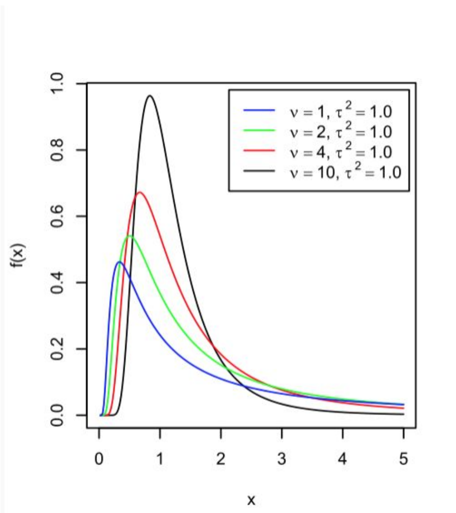
$$p(y|\mu, \sigma^2) = \frac{1}{y \cdot \sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(\log y - \mu)^2\right]$$

μ is assumed to be known = 3.5
$\sigma^2$ is unknown with a non-informative prior $p(\sigma^2) = 1 / \sigma^2$, resulting in the posterior for $\sigma^2$ is the Scaled Inv-$\chi^2$ distribution $Inv - \chi^2(n, \tau^2)$ where $\tau^2 = \frac{\sum_{i=1}^{n}(\log y_i - \mu)^2}{n}$.

**a)** Simulate 10 000 draws from the posterior $\sigma^2$:

```
tau2 <- function(..Y, my) {
  n <- length(..Y)
  return(sum((log(..Y)-my)^2)/n)
}

data <- c(14, 25, 45, 25, 30, 33, 19, 50, 34, 67)
my <- 3.5
n <- length(data)
nDraws = 10000

draw <- rchisq(n=nDraws, df=n)
sigma2 <- n*tau2(data, my)/draw
hist(sigma2, breaks=1000)
```



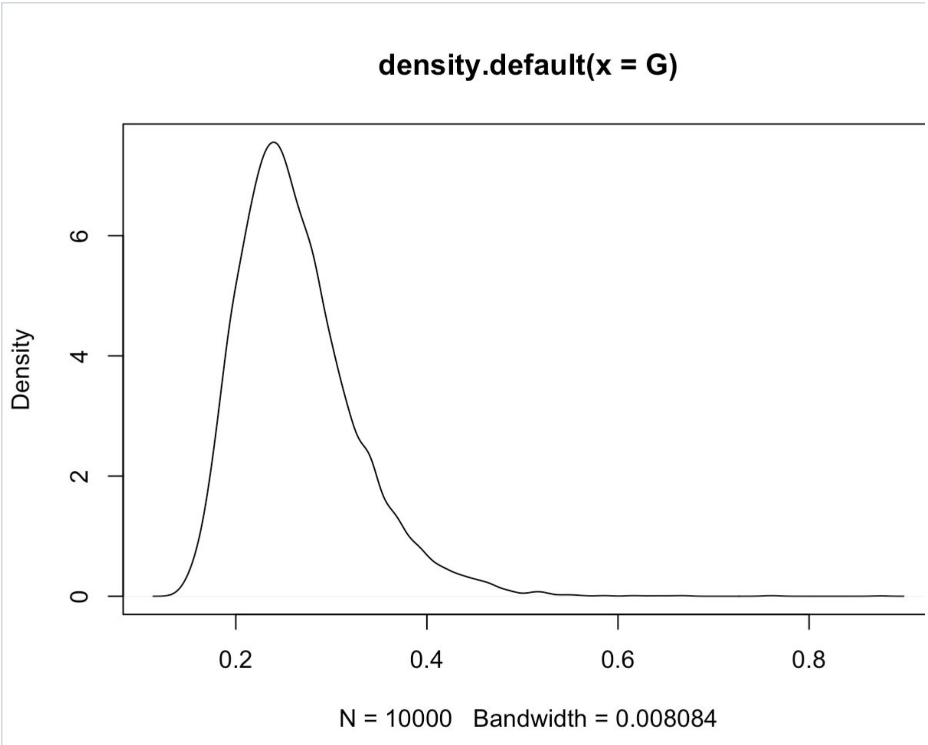




Compared to the theoretical $Inv - \chi^2(n, \tau^2)$ taken fromwikipedia it has a similar shape.

**b)** Using Gini coefficient to measure income inequality. Given that incomes follow a logN distribution then $G = 2\Phi\left(\sigma/\sqrt{2}\right) - 1$ where Φ(σ/sqrt(2)) is the probability of x <= σ/sqrt(2) in a N(0,1) distribution.

Using the posterior draws from a) we compute the posterior distribution of G for current data:

```
sigma = sqrt(sigma2)
G <- 2 * pnorm(sigma/sqrt(2), mean = 0, sd = 1) - 1
hist(G, breaks=100)
```

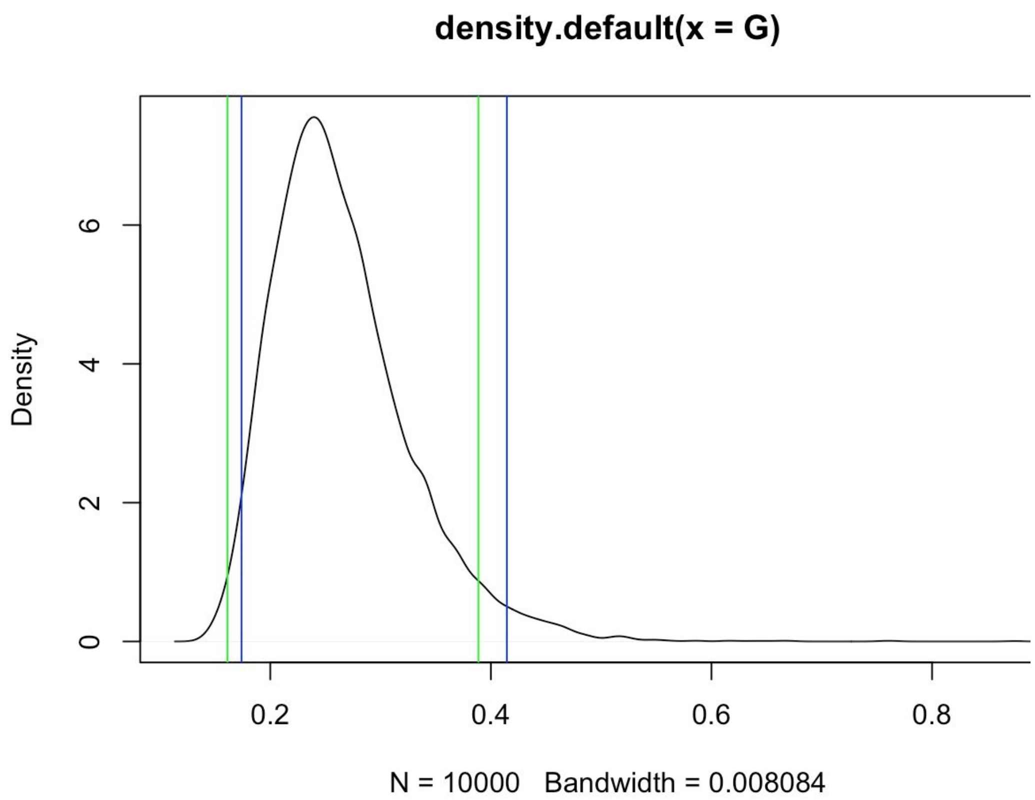


**c)** Computing the 95% equal tail credible interval for G using the posterior draws from b)

```
# equal tail
G.sorted <- sort(G)[(nDraws*0.025+1):(nDraws*0.975)]
G.credible_interval <- c(min(G.sorted), max(G.sorted)) # Credible interval
print(G.credible_interval) #0.1738928 0.4144967
```

Compute 95% highest posterior density interval:

```
# density
G.density <- density(G)
G.density.df <- data.frame(x = G.density$x, y = G.density$y)

G.density.ordered <- G.density.df[with(G.density.df, order(y)),]
G.density.ordered <- data.frame(x = G.density.ordered$x, y = G.density.ordered$y)
G.density.dn = cumsum(G.density.ordered$y)/sum(G.density.ordered$y)
G.0.05 = which(G.density.dn >= 0.05)[1]
G.density.ordered.95 = G.density.ordered[(G.0.05+1):512,]
G.density.interval <- c(min(G.density.ordered.95$x),max(G.density.ordered.95$x))
```




Where green depicts the HPD interval and blue depicts the credible interval. We can see that the HPD interval is closer to the mass of the posterior which seems like a more reasonable estimated interval result for G.

# 3. Bayeisan inference - von Mises distribution

**a)** Given observations y of modelled as a von Mises distribution and a exponential prior knowledge of the value of kappa):

(note: k = kappa)

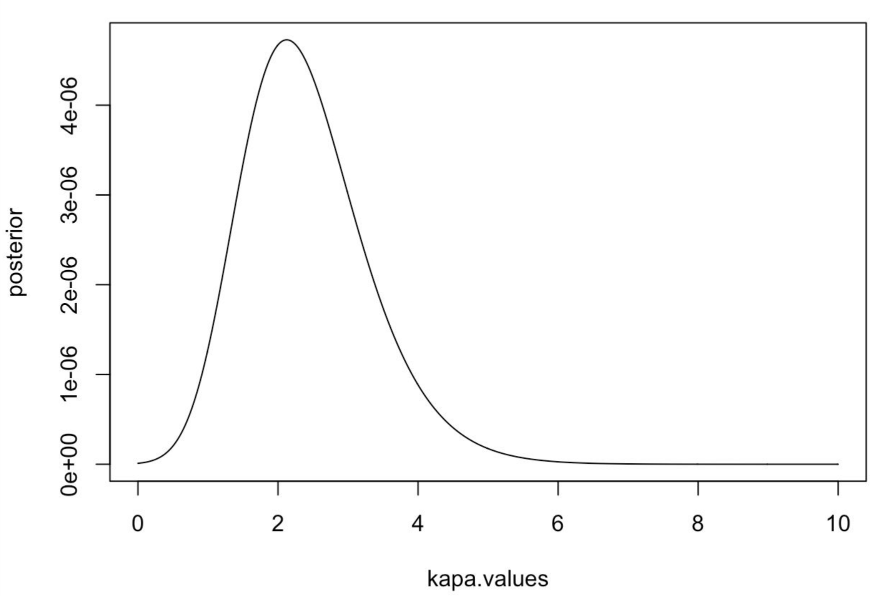
$$Prior: \ p(k) \ = \ \lambda e^{-\lambda k} = //\lambda \ = \ 1// \ = \ e^{-k}$$

$$Likelihood: \ p(Y \mid k, \ \mu) \ = \ p( \ Y \mid k) \ = \ // \ y \ independent \ // \ = \ \prod_{i=1}^{10} p(y \mid k \ ) \ = \prod_{i=1}^{10} \frac{exp\{k * cos \ ( \ y - \mu)\}}{2\pi I_0(k)}$$

$$Posterior: \ p(k|Y, \mu) \ = \ // \ \mu \ known \ // \ = \ p(k \mid Y) \ = \ p(Y \mid k) \ * \ p \ (k) \ = \ \prod_{i=1}^{10} \frac{exp\{k * cos \ ( \ y - \mu)\}}{2\pi I_0(k)} \ * \ e^{-k}$$

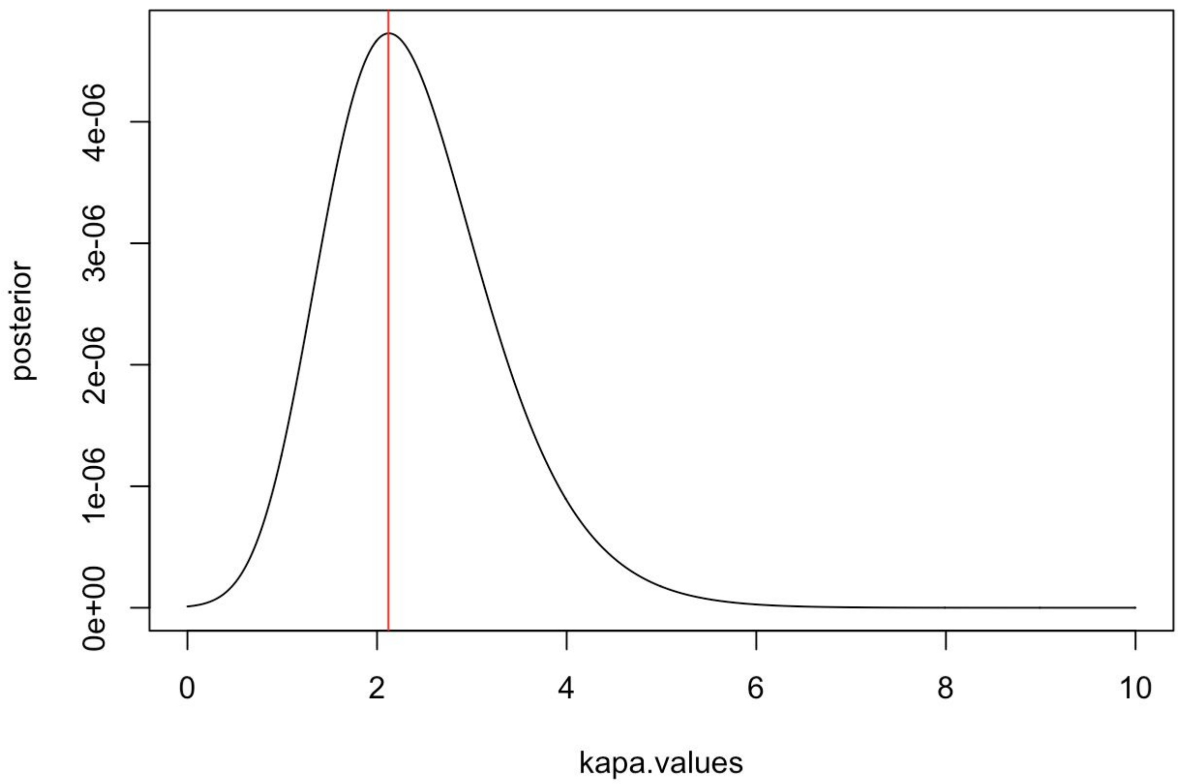
Plotting the posterior of kappa using a fine grid of kappa values:

```
likelihoodVonMises <- function(y, k, my) {
    return(prod(exp(k*cos(y-my))/(2*pi*besselI(k, nu=0))))
}

y.values <- c(-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.07, 2.02)
kapa.values <- seq(0, 10, 0.01)
my <- 2.39

# Vector of prior values, given kapas
kapa.prior <- dexp(kapa.values)

# Calculate likelihood for each kapa, given y-vector
likelihoods = numeric()
i = 1
for(k in kapa.values) {
  likelihoods[i] = likelihoodVonMises(y.values, k, my)
  i = i + 1
}

posterior = likelihoods * kapa.prior # Vector of posterior values
plot(kapa.values, posterior, type='l') # Plot
```



**b)**

Observing the graph from a) we find the approximate posterior mode of kappa taking the kappa resulting in the highest posterior probability for given y and kappa. Reading from the image around 2.10 seems reasonable:




```
# b)
max.posterior = which.max(posterior) # Find maximum posterior value
kapa.max.posterior = kapa.values[max.posterior] # Find maximum kapa value given max value in poster
# Max kapa: 2.12
```

Calculated from the grid values gives kappa = 2.12