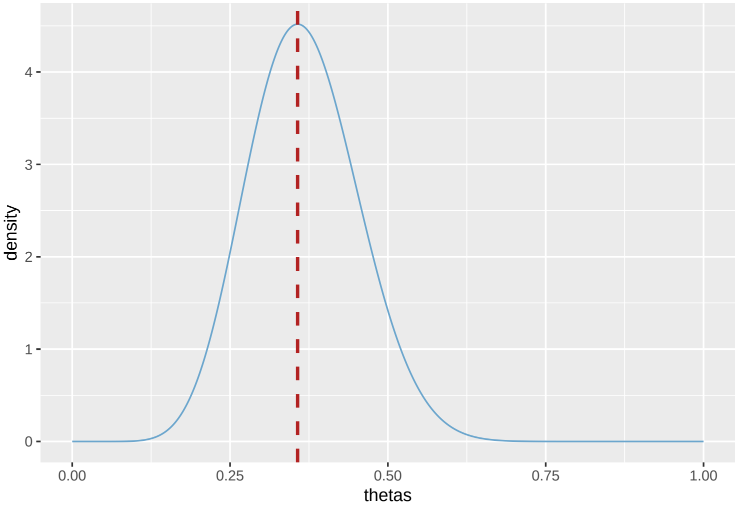
TDDE07 Bayesian Learning Lab1

Shahin Salehi & William Bergekrans

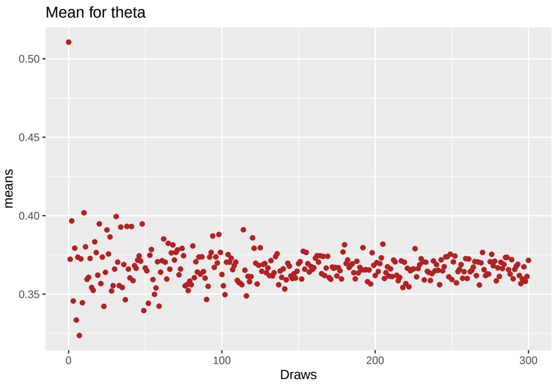
April 2020

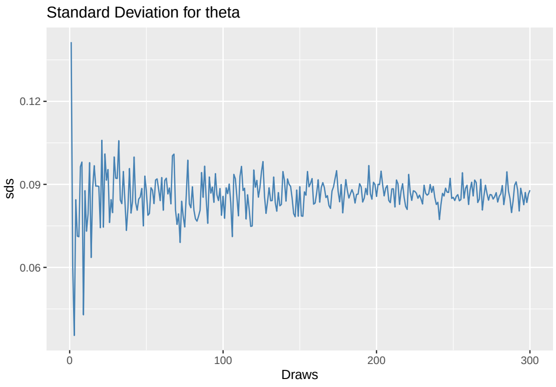
1. Daniel Bernoulli

Because we have $S = 8$ successes in $n = 24$ trials it means the number of failures is $f = 16$. Therefore the posterior distribution is $\theta|y \sim Beta(11, 19)$. The density distribution for θ is as follows:



1a) The true values of the $Beta(11, 19)$ distribution are 0,367 for the mean and 0,0865 for the standard deviation. We plot the mean and standard deviation for different number of draws.

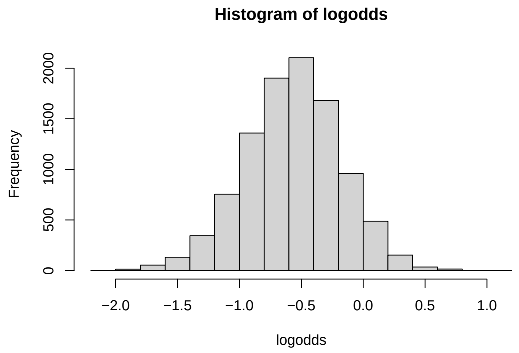

Mean for theta

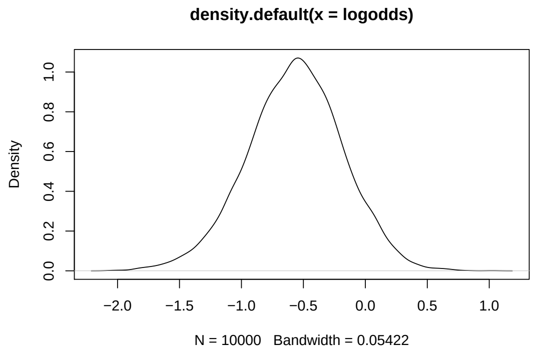

Standard Deviation for theta

Both graphs clearly converges towards the true values when the number of draws increases. This is to be expected as the values are drawn from a $Beta(11, 19)$ distribution on which the true values are based on.

1b) The number of draws is here set to 10 000. We calculate the posterior probability $Pr(\theta > 0.4|y)$. The observed probability using 10000 samples from the beta distribution is 0.3383 and the true probability calculated using the pbeta-function is 0.3426654.

1c) Here we use the log-odds function $\phi = \frac{\theta}{(1-\theta)}$ on the same sample used in the previous section 1b. The calculated values are visualized in the following histogram and density distribution graph.





From the density distribution graph we can see that there is a close similarity with the normal distribution. The expected value is the median x-value of the density function which is -0.513 and the standard deviation is 0.054.Therefore the posterior distribution is $\phi \sim N(-0.513, 0.00294)$.

Code for part 1 For the complete Rmd file see the other hand-in file.

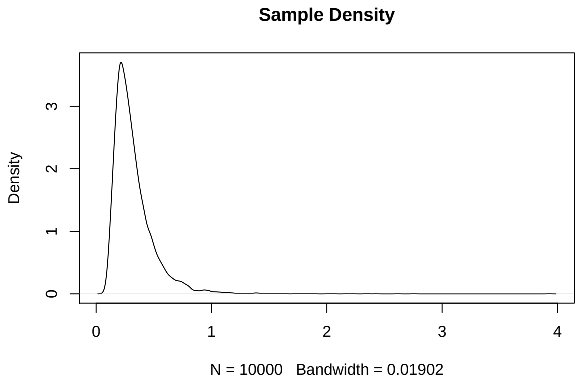
```
# Code for a.
means <- rep(0,301)
sds <- rep(0,301)
# 1a Mean and SD for different number of draws
for (draw in 0:301) {
  # Draw n number of observations from the beta distribution.
  sample <- rbeta(draw,11,19)
  means[draw] <- mean(sample) # Calculate mean
  sds[draw] <- sd(sample) # Calculate standard deviation
}

# Plot the means and standard deviation
Draws <- seq(0,300,1)

ggplot(data.frame(means), aes(x=Draws)) +
  geom_point(aes(y=means), color="firebrick") +
  ggtitle("Mean for theta")

ggplot(data.frame(means), aes(x=Draws)) +
  geom_line(aes(y=sds), color="steelblue") +
  ggtitle("Standard Deviation for theta")

# Code for 1b.
# Generate points from a beta(11,19) distribution.
set.seed(12345)
sample <- rbeta(10000, 11,19)

prob <- length(sample[sample > 0.4]) / length(sample)

true_prob <- pbeta(0.4,11,19, lower.tail = FALSE)
```

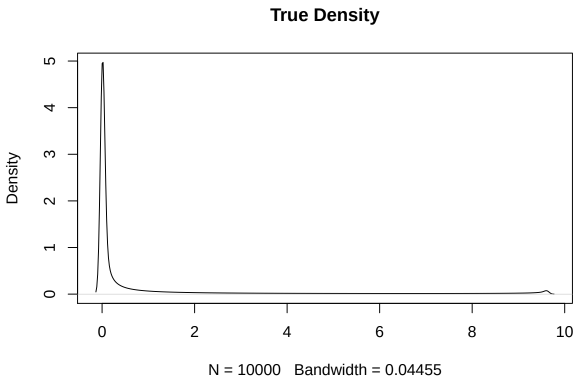
```
# Code for 1c.
# use the same sample from the beta(11,19) distribution that was used in 1b.
logodds <- log(sample/(1-sample))

hist(logodds) # Print the log-odds density as a histogram.

dens <- density(logodds)
plot(dens)
var <- 0.05422^2 # variance from standard deviation
```

2. Log-normal distribution and the Gini coefficient

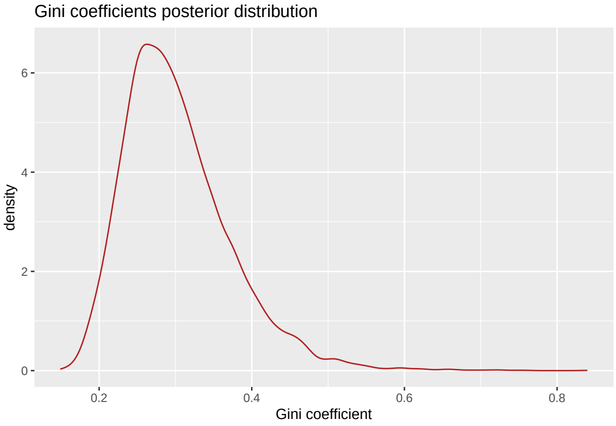
The log-normal distribution used in this case has a known mean of $\mu = 3.8$ and unknown variation σ^2. The prior to be used is the non-informative prior $p(\sigma^2) \propto \frac{1}{\sigma^2}$.

2a) First a sample of 10 000 observations is sampled from the posterior distribution for $\sigma^2|x \sim \text{Inv -}\chi^2(n, \tau^2)$ where $\tau^2 = \frac{\sum_{i=1}^{2}(logy_i - \mu)^2}{n}$. In the following two graphs we see the true and sampled probability densities for different values of σ^2.

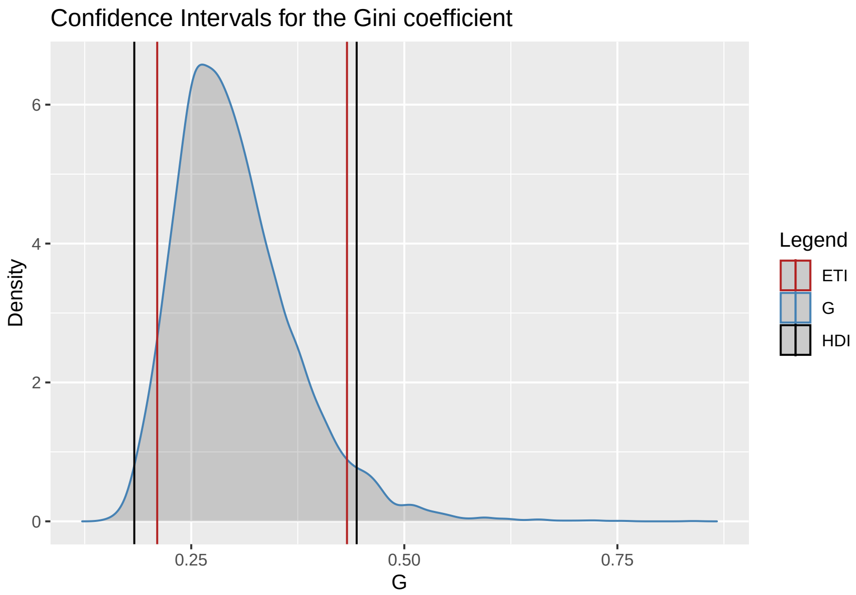




2b) Now the gini-coefficient is calculated for every value in our sample of 10.000 observations from 2a. The posterior density function for the Gini coefficient based on our samples is visualized in the following graph:



2c) Now we want to compute a 90 percent tail credible interval (a, b) for the Gini coefficient G. We calculate the interval using the posterior draw for the gini coefficient made in part 2b above.



When the two intervals are compared we see that the highest probability denstiy interval is wider than the equal tailed interval.

Code for part 2 The following code was used for part 2a-c.

```
# Question 2
# Code for 2a
incomes <- c(38, 20, 49, 58, 31, 70, 18, 56, 25, 78)
u <- 3.8 # Mean values
n <- length(incomes) # Degrees of freedom
sampleSize <- 10000 # Size of sample

# Calculate tao-squared to be used in the posterior.
t2 <- sum((log(incomes)-u)^2) / n

set.seed(12345)
sample <- rinvchisq(sampleSize, n, t2)
sampleDen <- density(sample)

set.seed(12345)
truePost <- density(dinvchisq(seq(0.0001,1,0.0001), n))

plot(sampleDen, main = "Sample Density")
plot(truePost, main = "True Density")

# Code for 2b
gini <- 2* pnorm(sqrt(sample)/sqrt(2), 0, 1) - 1

frame <- data.frame(Gini = gini)

ggplot(frame, aes(x=Gini)) +
  geom_density(color = "firebrick") +
  ggtitle("Gini coefficients posterior distribution") +
  xlab("Gini coefficient")
```

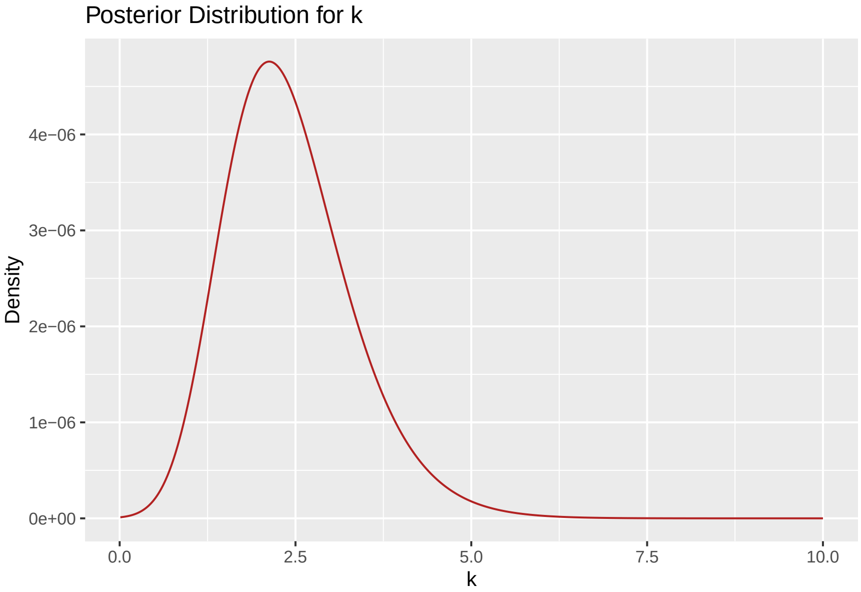
```
# Code for 2c
ci_equal <- ci(gini, ci=0.9, method="ETI")
g_den <- density(gini)

ci_hdi <- hdi(g_den, ci=0.9)

frame <- data.frame(
  # Density graph
  x <- g_den$x,
  y <- g_den$y,
  # Equal tail interval
  cie_low <- ci_equal$CI_low,
  cie_high <- ci_equal$CI_high,
  # Highest prob density interval
  cih_low <- ci_hdi[1],
  cih_high <- ci_hdi[2]
)
colors = c("ETI"="firebrick", "G"="steelblue", "HDI"="black")
ggplot(frame, aes(x,y)) +
  geom_area(aes(color="G"), alpha=0.2) +
  geom_vline(aes(xintercept = cie_low, color="ETI")) +
  geom_vline(aes(xintercept = cie_high, color="ETI")) +
  geom_vline(aes(xintercept = cih_low, color="HDI")) +
  geom_vline(aes(xintercept = cih_high, color="HDI")) +
  ggtitle("Confidence Intervals for the Gini coefficient") +
  labs(x = "G",
       y = "Density",
       color = "Legend") +
  scale_color_manual(values= colors)
```

3. Bayesian inference in the von Mises distribution

We have 10 wind direction observations from a von Mises distribution. The observations therefore follow:

$$p(y|\mu,k) = \frac{exp[k * cos(y-\mu)]}{2\pi I_0(k)}, -\pi \leq y \leq \pi$$

$I_0(k)$ is the modified Bessel function of first kind of order zero. μ is the mean direction and k $>$ 0 is the concentration parameter. We assume that μ is 2.39. The posterior distribution for k is seen in the following plot:



3b) The mode of k with the highest probability.

The mode is 2.13.

Code) Code for part 3a-b.

```r
# Part 3
# Code for 3a
wind <- c(40, 303, 326, 285, 296, 314, 20, 308, 299, 296)
wind_rad <- c(-2.44, 2.14, 2.54, 1.83, 2.02, 2.33, -2.79, 2.23, 2.08, 2.02)
u <- 2.39 # Given constant
ks <- seq(0.01, 10, 0.01)
prior <- dexp(ks) # Caclucalte the exponential prior
i = 1
likelihood <- numeric(length(ks))
for(k in ks) {
  likelihood[i] <- prod(exp(k*cos(wind_rad-u)) / (2*pi*besselI(k, nu=0)))
  i = i+1
}
posterior <- prior * likelihood
frame <- data.frame(Density = posterior, k = ks)
ggplot(frame, aes(k, Density)) +
  geom_line(color="firebrick") +
  ggtitle("Posterior Distribution for k")

# Code for 3b
m <- ks[which.max(posterior)]
```