Sampling Distribution of an Estimator

For the Beta Binomial distribution, simulate the sampling distribution of the MLE. Compare this distribution to the theoretical distribution from large sample theory.

```
library(ggplot2)
library(mvtnorm)
```

Functions to compute the density and draw from Beta Binomial (BB) distribution.

```
d.beta.binom <- function(x, Pi, rho, m, log = FALSE)
{
    a <- Pi * rho^(-2) * (1 - rho^2)
    b <- (1 - Pi) * rho^(-2) * (1 - rho^2)
    log.ff <- lgamma(m + 1) - lgamma(x + 1) - lgamma(m - x + 1) +
        lgamma(a + x) + lgamma(b + m - x) - lgamma(a + b + m) +
        lgamma(a + b) - lgamma(a) - lgamma(b)
    if (log) return(log.ff)
    else return(exp(log.ff))
}

r.beta.binom <- function(n, Pi, rho, m)
{
    a <- Pi * rho^(-2) * (1 - rho^2)
    b <- (1 - Pi) * rho^(-2) * (1 - rho^2)
    z <- rbeta(n, a, b)
    rbinom(n, size = m, prob = z)
}
```

Function to compute the MLE via `optim`.

```
mle <- function(y, m, par.init = c(0, 0))
{
    loglik <- function(par) {
        Pi <- plogis(par[1])
        rho <- plogis(par[2])
        sum(d.beta.binom(y, Pi, rho, m, log = TRUE))
    }
    optim(par.init, loglik, method = "L-BFGS-B", control = list(fnscale = -1))
}
```

Example of drawing a dataset from BB and fitting the model.

```
set.seed(1234)
n <- 200
m <- rep(20, n)
Pi.true <- 0.6
rho.true <- 0.3
y <- r.beta.binom(n, Pi.true, rho.true, m)

par.hat <- mle(y, m)$par
Pi.hat <- plogis(par.hat[1])
rho.hat <- plogis(par.hat[2])

print(Pi.hat)
```

```
## [1] 0.6049245
```

```
print(rho.hat)
```

```
## [1] 0.273484
```

Simulation to find the distribution of the MLE.

```r
set.seed(1234)
n <- 200
m <- rep(20, n)
Pi.true <- 0.6
rho.true <- 0.3

R <- 5000
res <- matrix(NA, R, 2)

for (r in 1:R) {
    y <- r.beta.binom(n, Pi.true, rho.true, m)
    par.hat <- mle(y, m)$par
    res[r,1] <- plogis(par.hat[1])
    res[r,2] <- plogis(par.hat[2])
}
```

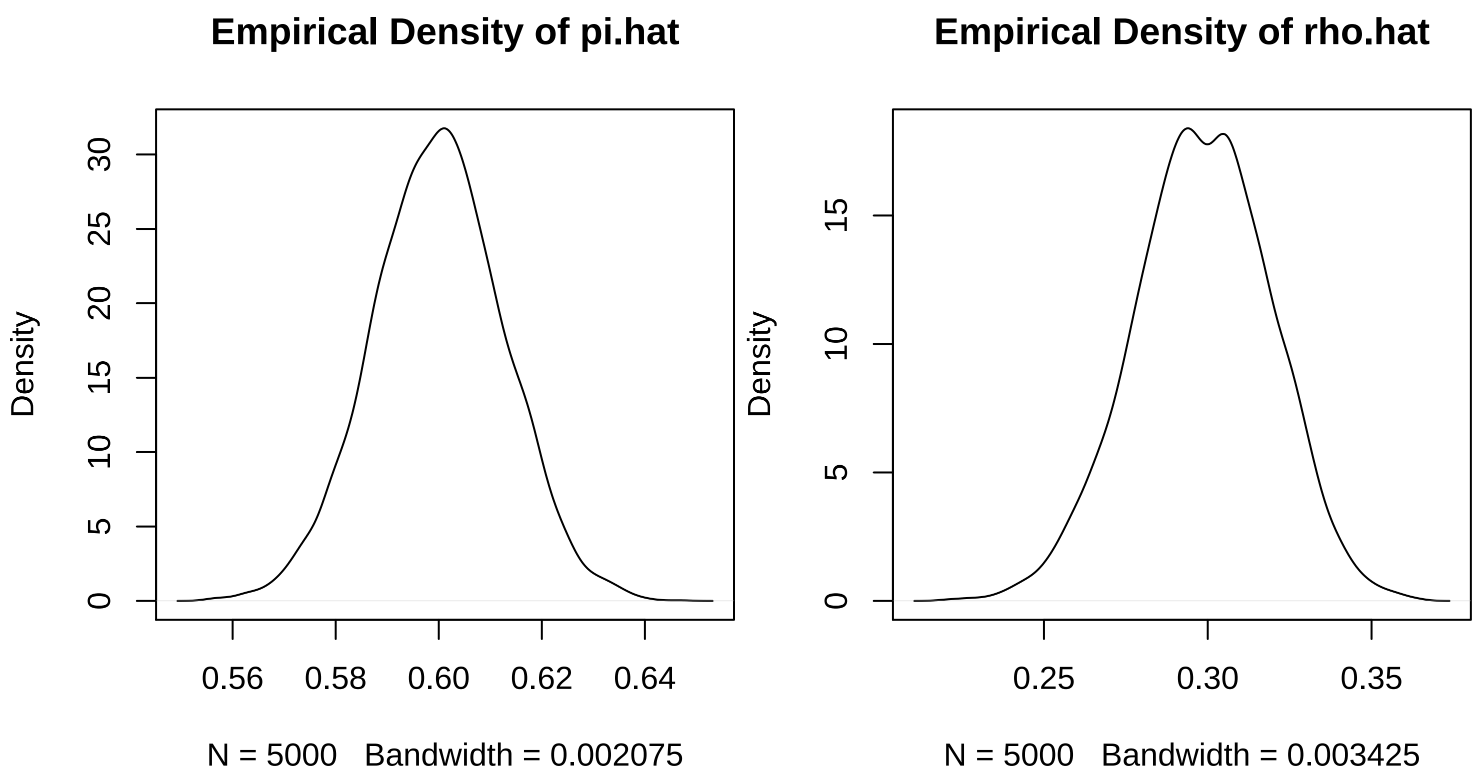
```r
mu.avg <- colMeans(res)
Sigma.avg <- var(res)
print(mu.avg)
```

```
## [1] 0.5998992 0.2983611
```

```r
print(Sigma.avg)
```

```
##               [,1]          [,2]
## [1,]  1.610243e-04 -2.010651e-05
## [2,] -2.010651e-05  4.368686e-04
```

```r
# Univariate plots of empirical density
plot(density(res[,1]), main = "Empirical Density of pi.hat")
plot(density(res[,2]), main = "Empirical Density of rho.hat")
```



Plot of 2-d empirical density from simulation versus the normal distribution from large sample theory.

```r
dat.sim <- data.frame(x = res[,1], y = res[,2])

dat.mvn <- expand.grid(
    x = seq(min(dat.sim$x), max(dat.sim$x), length.out = 100),
    y = seq(min(dat.sim$y), max(dat.sim$y), length.out = 100))
```

```r
dat.mvn$dens <- dmvnorm(dat.mvn, mean = mu.avg, sigma = Sigma.avg)

ggplot(dat.sim, aes(x=x, y=y)) +
    stat_density_2d(geom = "raster", aes(fill = ..density..), contour = FALSE) +
    scale_fill_gradient(low="lightcyan", high="purple") +
    ggtitle("Empirical Density of MLE")

ggplot(dat.mvn, aes(x=x, y=y)) +
    geom_raster(aes(fill = dens)) +
    scale_fill_gradient(low="lightcyan", high="purple") +
    ggtitle("MVN Density Based on MLE")
```