Monte Carlo Estimation

May 1, 2018

Parctical virtue: The practical virtue of simulation methods in general, including MCMC, is that, given a set of random draws $\theta_1, \theta_2, ..., \theta_n$ from the posterior distribution, one can estimate virtually all summaries of interest from the posterior distribution directly from the simulations.

For example, means, variances, and posterior intervals for a quantity of interest $h(\theta)$ can be estimated using the sample mean, variance, and central intervals of the values $h(\theta_1), h(\theta_2), ..., h(\theta_n)$. MCMC methods have been successful because they allow one to draw simulations from a wide range of distributions, including many that arise in statistical work, for which simulation methods were previously much more diffcult to implement.

1 Monte Carlo Integration

Monte Carlo estimation refers to simulating hypothetical draws from a probability distribution. In order to calculate important quantities of that distribution. Some of these quantities might include the mean, the variance, the probability of some event, or the quantiles of the distribution.

Let's suppose:

$$\theta \sim G(a = 2, b = \frac{1}{3})$$

Let's calculate the expected value :

$$E(\theta) = \int_0^\infty \theta p(\theta) d\theta = \int_0^\infty \theta \frac{b^a}{\Gamma(a)} \theta^{a-1} e^{-b\theta} d\theta = \frac{a}{b}$$

We can try retrieving the same using Monte Carlo Estimation, so we generate θ^*:

$$\theta_i^* \ i = 1, ..., m$$

By the Central Limit Theorem we derive:

$$\bar{\theta^*} = \frac{1}{m} \sum_{i=1}^{m} \theta_i^* \approx N(\mu, \sigma^2)$$

What about the variance?

$$Var(\theta) = \int_0^\infty (\theta - E(\theta))^2 p(\theta) d\theta$$

With Monte Carlo Estimation:

This method of Monte Carlo estimation can be use to calculate many different integrals. Let $h(\theta)$, and we want to calculate:

$$\int h(\theta)p(\theta)d\theta = E[h(\theta)] \approx \frac{1}{m}\sum_{i=1}^{m} h(\theta_i^*)$$

So we would apply this function h to every simulated sample and then take the average of the results.

Indicator function

Let $h(\theta) = I_{\theta<5}(\theta)$:

$$E(h(\theta)) = \int_0^\infty I_{\theta<5}(\theta)p(\theta)d\theta$$

$$= \int_0^5 1p(\theta)d\theta + \int_5^\infty 0p(\theta)d\theta$$

$$= Pr[0 < \theta < 5]$$

What does that mean?

It means that we can approximate this probability right here by drawing many samples, θ_i^*.And:

$$\approx \frac{1}{m}\sum_{i=1}^{m} I_{\theta^*<5}(\theta_i^*)$$

What this function does is simply counts how many of our simulated values meet this criteria. And then divides by the total number of samples taken. So this approximates the probability that theta is less than five, that's pretty convenient.

Likewise, we can approximate quantiles of a distribution. If we're looking for a value z, that makes it so that the probability of being less than z is 0.9 for example. We would simply arrange the samples theta star I in ascending order. And then we would find the smallest value of the theta stars that's greater than 90% of the others. In other words we would take the 90th percentile of the theta stars to approximate the 0.9 quantile of the distribution.

2 Monte Carlo error and marginalization

How good is an approximation by Monte Carlo sampling? Again, we can turn to the central limit theorem, which tells us that the variance of our estimate is controlled in part by m, our sample size. If we want a better estimate, we need to choose a larger end value.

Example

Let $\bar{\theta^*} \dot{\sim} N(E(\theta), \frac{Var(\theta)}{m})$

$$Va\hat{r}(\theta) = \frac{1}{m}\sum_{i=1}^{m}(\theta^* - \bar{\theta^*})^2$$

$$Standard\ Error = \sqrt{\frac{Va\hat{r}(\theta)}{m}}$$

We can also obtain Monte Carlo samples from hierarchical models.

** Example - Hierarchical Models**

Let a binomial random variable y. So:

$$y|\phi \sim Bin(10, \phi)$$
$$\phi \sim Beta(2, 2)$$

Given any hierarchical model, we can always write out the joint distribution of y and ϕ.

$$p(y, \phi) = p(\phi)p(y|\phi)$$

To simulate from this joint distribution, we're going to repeat the following steps for a large number of samples, m.

1. ϕ_i^* from Beta
2. Given Φ_i^* draw $y_i^* \sim Bin(10, \Phi_i^*)$

To simulate from this joint distribution, we're going to repeat the following steps for a large number of samples, m. (y_i, Φ_i)

These pairs right here are drawn from their joint distribution.

One major advantage of Monte Carlo simulation is that marginalizing these distributions is easy. Calculating the marginal distribution of y might be difficult here. It would require that we integrate this expression with respect to ϕ, to integrate out the ϕs.

3 Monte Carlo Estimation Examples in R

Now that we've set the random seed for our session, let's start with an example from the previous segment. Where our random variable $\theta \sim Gamma(a = 2, b = \frac{1}{3})$. This gamma distribution could represent the posterior distribution of θ if our data came from a Poisson distribution with mean θ and we had used a conjugate gamma prior.

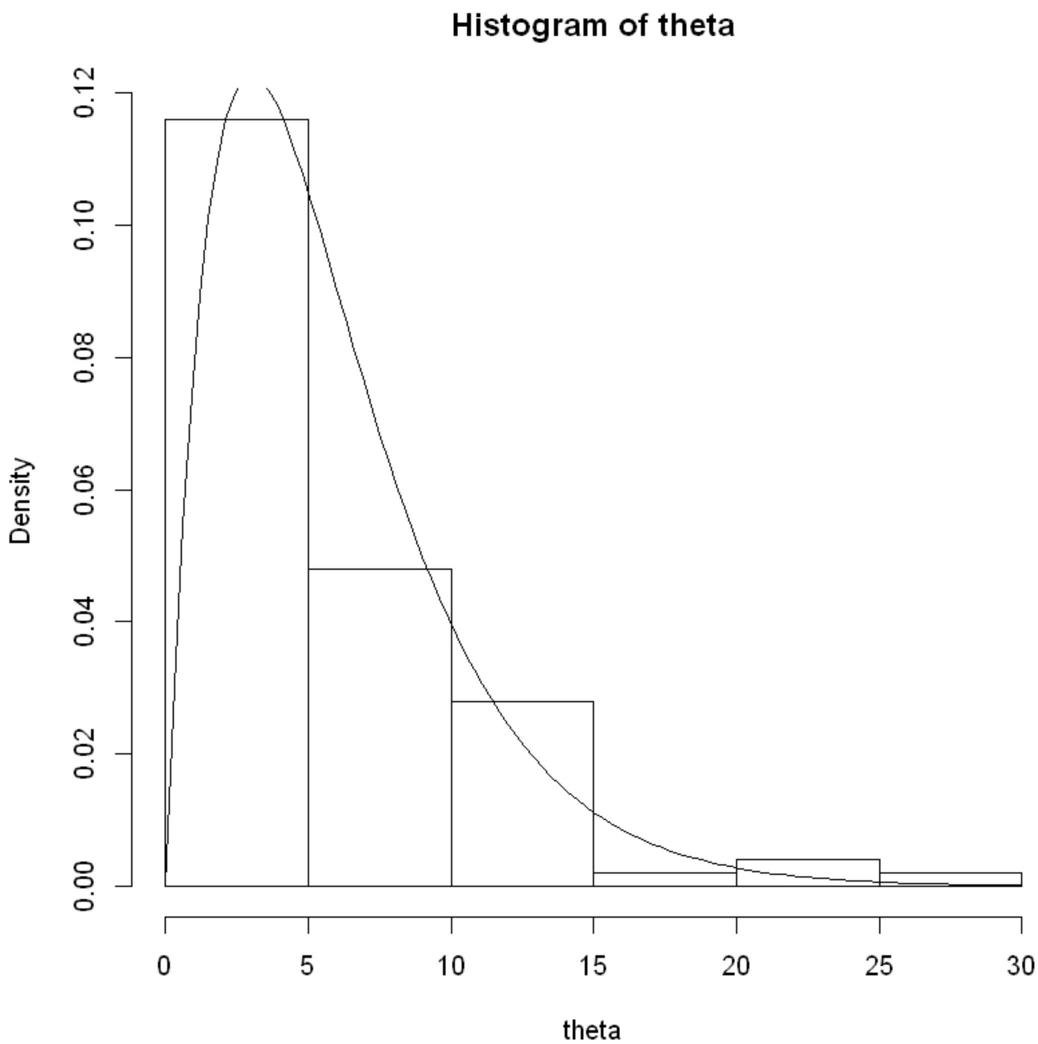
Let's start with Monte Carlo sample size 100.

```
In [1]: set.seed(1)

        m <- 100
        a <- 2
        b <- 1/3
```

```
In [5]: # To simulate the values we use rgamma.
        theta <- rgamma(n = m, shape = a, rate = b)

        # Let's take a look what we generated, we also set freq to F so that
        # it returns probability and not counts

        hist(theta, freq = F)

        # Let's compare this to the theoretical probability density function
        #for the gamma distribution

        curve(dgamma(x, shape = a, rate = b), colour = "blue", add = T)
```

```
# data are pretty good
```

```
Warning message in plot.xy(xy.coords(x, y), type = type, ...):
""colour" is not a graphical parameter"
```



In [6]:
```
# Let's find a Monte Carlo approximation to the expected value of theta.

sum(theta)/m

#OR

mean(theta)
```

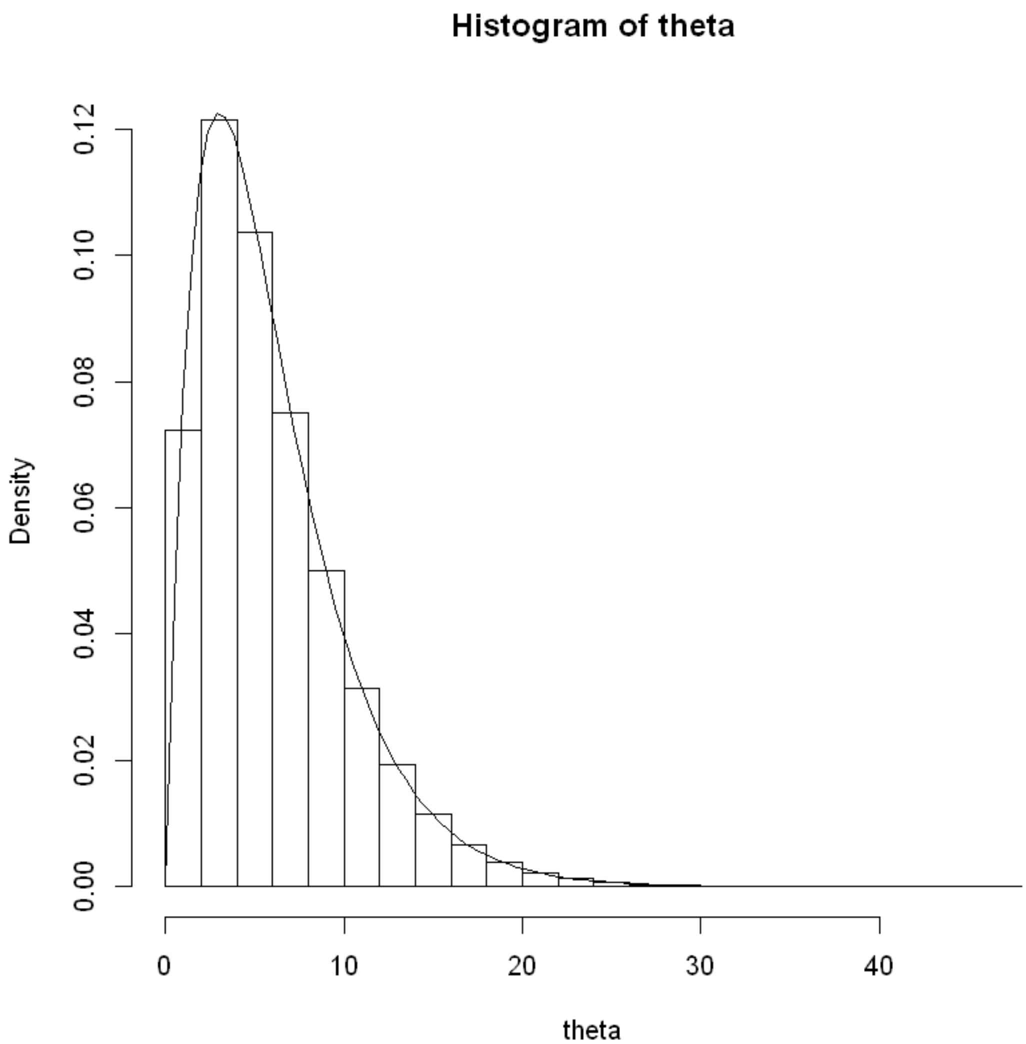
```
# How does this compare with the true value of the expected value theta?

a/b
```

5.87656407421684
5.87656407421684
6

```
In [11]: # Pretty good, but let's increase m to see if there is any improvement

         m <- 100000
         theta <- rgamma(n = m, shape = a, rate = b)
         hist(theta, freq = F)
         curve(dgamma(x, shape = a, rate = b), colour = "blue", add = T)
         mean(theta)

         # Pretty cool
```

```
Warning message in plot.xy(xy.coords(x, y), type = type, ...):
""colour" is not a graphical parameter"
```

5.99695679720809




```
In [12]: # What about variance?

         var(theta)

         a/b^2

         #Pretty close, right?
```

18.0916551104863
18

We can also use our Monte Carlo simulation to approximate other characteristics of this gamma distribution.

For example, if we want to approximate the probability that $\theta < 5$. We can simply count how many of our simulated θs are less than 5. Let's do this.

In [17]:
```
# First create an indicator variable

ind <- theta < 5

# Let's calculate the approximation to the probability
# that theta is less than 5
mean(ind)

#Let's compare this to the true probability

pgamma(q = 5, shape = a, rate = b)


#what about the 90th percentile of this distribution.
quantile(theta, probs = 0.9)

#What about the actual true quantile?
qgamma(p = 0.9, shape = a, rate = b)
```

0.49675
0.496331725766502
90\%: 11.7082278237797
11.6691605096023

4 Monte Carlo Error in R

We can use the central limit theorem to approximate how accurate our Monte Carlo estimates are.

To create a standard error for our Monte Carlo estimate, we will use the sample standard deviation divided by the square root of m.

In [20]:
```
se = sd(theta)/sqrt(m)

#Let's create a confidence interval for our Monte Carlo approximation.
#two times the standard error which is, we are reasonably confident,
#about 95% confident.

#That the Monte Carlo estimate for the expected value of theta is no
#more than this far from the true value of the expected value of theta.

2*se

# The intervals are
mean(theta)-2*se; mean(theta) + 2*se
```

0.0269010446715263
5.97005575253656
6.02385784187962

In [21]:
```
#The same applies for other Monte Carlo estimates.
#For example
se = sd(ind)/sqrt(m)
2*se
#it looks like, in this case, our Monte Carlo estimate is within 0.003
#of the true value.
```

0.00316222666752005

In [24]:
```
#Let's also do the second example where we simulate from a hierarchical model.

#1 simulate phi_i from Beta(2,2)
#2 simulate y_i from Binom(10, phi_i)

m <- 1e5
y <- numeric(m)
head(y)
phi <- numeric(m)

# To implement our algorithm for this joint simulation

for (i in 1:m){
    # we first simulate the value phi from a beta distribution
    # we'll take a sigle draw with a = 2 and b = 2.
    phi[i] <- rbeta(1, shape1 = 2, shape2 = 2)
    # Now given that draw for phi we can simulate a draw for y.
    # The ith simulation come from a binomial distribution
    y[i] <- rbinom(1, size = 10, prob = phi[i])
}
```

1. 0 2. 0 3. 0 4. 0 5. 0 6. 0

In [25]:
```
# It's best to avoid using loops
# Let's do that in a vectorized form (more compacted and efficient)

phi <- rbeta(m, shape1 = 2, shape2 = 2)
y <- rbinom(m, size = 10, prob = phi)
```

If we are interested only in the marginal distribution of y, we can just ignore the draws for the phis. And treat the draws of y as a sample of its marginal distribution.

That distribution will not be a binomial distribution, it'll actually be a beta binomial distribution.

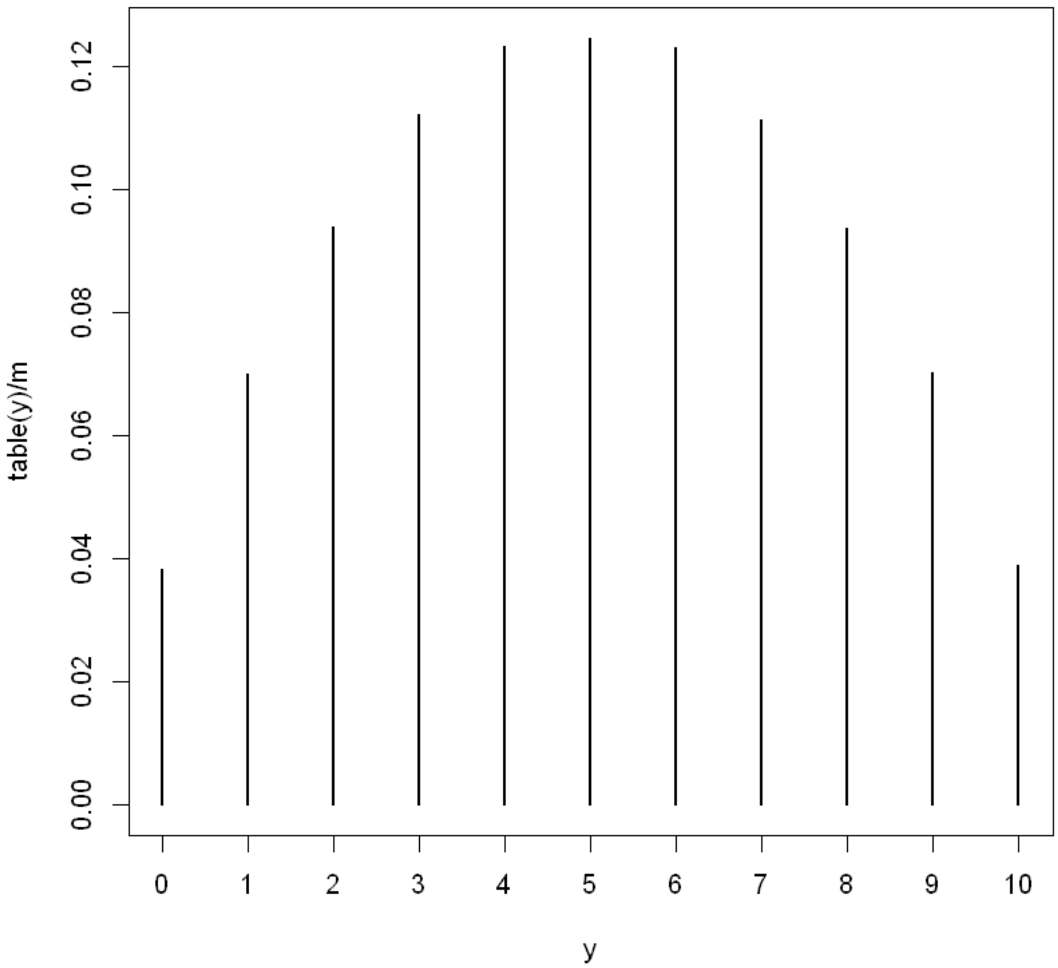
Conditional on phi, y follows a binomial distribution. But unconditionally, the marginal distribution of y is not binomial. Let's take a look at that distribution. First, we can do a table of the values of y. This will tell us how often each of the different values of y were drawn in our simulation. So if we run this table we can see how many of the simulations resulted in seven successes.

In [27]:
```
table(y)

# Let's approximate the probabilities

table(y)/m
```

```
y
    0     1     2     3     4     5     6     7     8     9    10
 3834  7015  9402 11225 12327 12468 12307 11130  9375  7030  3887


y
      0       1       2       3       4       5       6       7       8       9
0.03834 0.07015 0.09402 0.11225 0.12327 0.12468 0.12307 0.11130 0.09375 0.07030
     10
0.03887
```

In [29]:
```
#let's plot it
plot(table(y)/m)
# It is obviously a beta distribution

mean(y)
```

5.00034



5 Review

Question 1

If a random variable X follows a standard uniform distribution (X∼Unif(0,1)), then the PDF of X is p(x)=1 for 0≤x≤1.

We can use Monte Carlo simulation of X to approximate the following integral: $\int_0^1 x^2 dx = \int_0^1 x^2 \cdot 1 dx = \int \int_0^1 x^2 \cdot p(x) dx = E(X^2)$.

If we simulate 1000 independent samples from the standard uniform distribution and call them xi∗ for i=1,. . . ,1000, which of the following calculations will approximate the integral above?

a. $\frac{1}{1000} \sum_{i=1}^{1000} (x_i^* - \bar{x^*})^2$ where $\bar{x^*}$ is the calculated average of the x_i^* samples.

b. $\frac{1}{1000}\sum_{i=1}^{1000}(x_i^*)^2$

c. $\frac{1}{1000}\sum_{i=1}^{1000}x_i^*$

d. $(\frac{1}{1000}\sum_{i=1}^{1000}x_i^*)^2$

Answer 1 b.
Question 2
Suppose we simulate 1000 samples from a Unif(0,π) distribution (which has PDF p(x)=1π for 0≤x≤π) and call the samples xi∗ for i=1,. . . ,1000.

If we use these samples to calculate $\frac{1}{1000}\sum_{i=1}^{1000}\sin(x_i^*)$, what integral are we approximating?

a. $\int_0^1 \sin(x)dx$

b. $\int_0^1 \frac{\sin(x)}{\pi}dx$

c. $\int_0^{\pi} \frac{\sin(x)}{\pi}dx$

d. a. $\int_{-\infty}^{\infty} \sin(x)dx$

Answer 2

c.

Question 3
Suppose random variables X and Y have a joint probability distribution p(X,Y). Suppose we simulate 1000 samples from this distribution, which gives us 1000 (x_i^*, y_i^*) pairs.

If we count how many of these pairs satisfy the condition $x_i^* < y_i^*$ and divide the result by 1000, what quantity are we approximating via Monte Carlo simulation?

a. $Pr[X < Y]$

b. $Pr[E(X) < E(Y)]$

c. $Pr[X < E(Y)]$

d. $E(XY)$

Answer 3

a.

Question 4
If we simulate 100 samples from a Gamma(2,1) distribution, what is the approximate distribution of the sample average x∗¯=$\frac{1}{100}\sum_{i=1}^{100}\sin(x_i^*)$?

Hint: the mean and variance of a Gamma(a,b) random variable are $\frac{a}{b}$ and $\frac{a}{b^2}$ respectively.

a. Gamma(2,1)

b. Gamma(2,0.01)

c. N(2,0.02)

d. N(2,2)

Answer 4

c.

Due to the central limit theorem, the approximating distribution is normal with mean equal to the mean of the original variable, and with variance equal to the variance of the original variable divided by the sample size.

Question 5

Laura keeps record of her loan applications and performs a Bayesian analysis of her success rate θ. Her analysis yields a Beta(5,3) posterior distribution for θ.

The posterior mean for θ is equal to $\frac{5}{5+3} = 0.625$. However, Laura likes to think in terms of the odds of succeeding, defined as $\frac{\theta}{1-\theta}$, the probability of success divided by the probability of failure.

Use R to simulate a large number of samples (more than 10,000) from the posterior distribution for θ and use these samples to approximate the posterior mean for Laura's odds of success (E(θ1$-\theta$)).

Report your answer to at least one decimal place.

Answer 5

```
In [36]: m <- 1000
         a <- 5
         b <- 3
         theta <- rbeta(n = m, shape1 = a, shape2 = b)
         round(mean(theta/(1-theta)),1)
```

2.5

Question 6

Laura also wants to know the posterior probability that her odds of success on loan applications is greater than 1.0 (in other words, better than 50:50 odds).

Use your Monte Carlo sample from the distribution of θ to approximate the probability that $\frac{\theta}{1-\theta}$ is greater than 1.0.

Report your answer to at least two decimal places.

Answer 6

```
In [39]: ind <- theta/(1 - theta) > 1
         round(mean(ind),2)
```

0.78

Question 7

Use a (large) Monte Carlo sample to approximate the 0.3 quantile of the standard normal distribution (N(0,1)), the number such that the probability of being less than it is 0.3.

Use the quantile function in R. You can of course check your answer using the qnorm function.

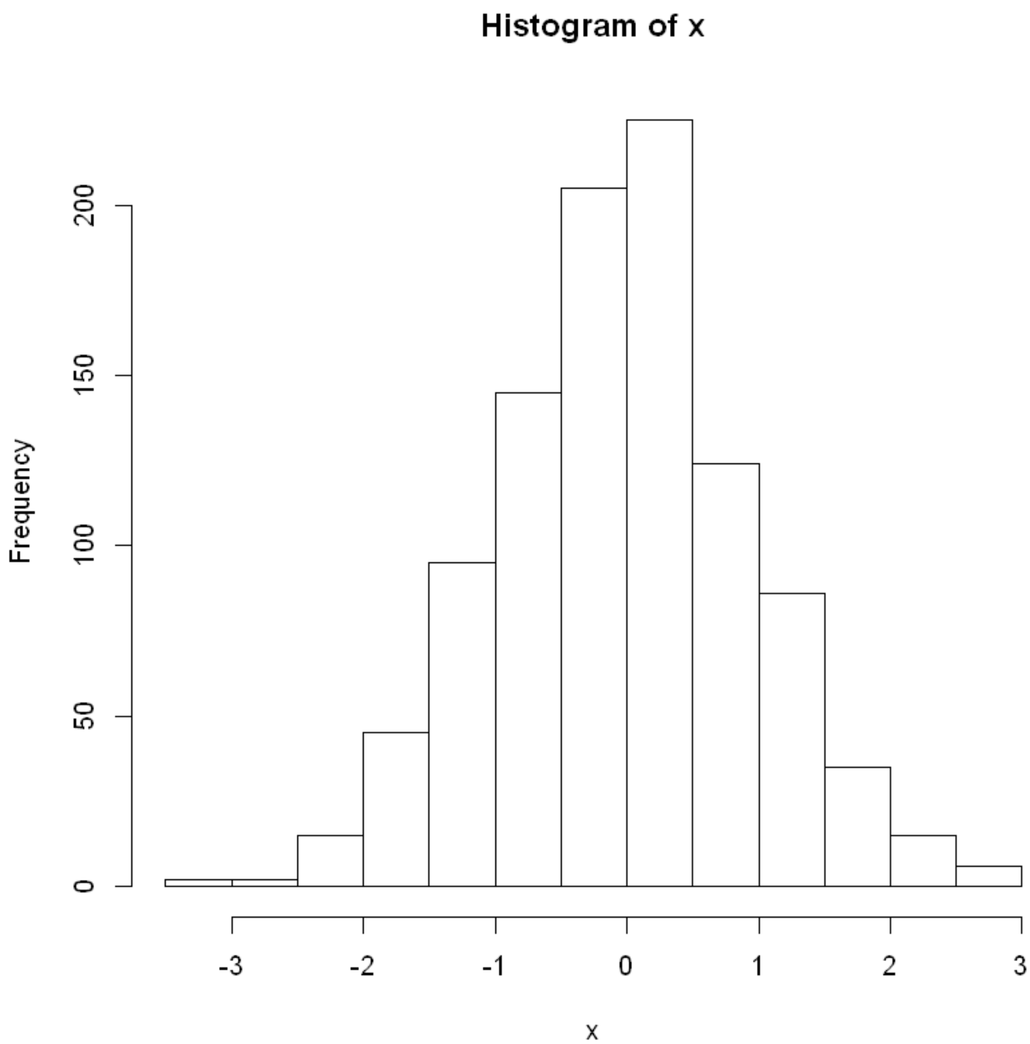
Report your answer to at least two decimal places.

Answer 7

In [50]:
```
mu <- 0
sigma <- 1
n <- 1000

x <- rnorm(n, mu, sigma^2)
#what about the 90th percentile of this distribution.
quantile(x, probs = 0.3)

qnorm(0.3, mu, sigma)
hist(x)
```

30\%: -0.523599781006521

-0.524400512708041



** Question 8**

To measure how accurate our Monte Carlo approximations are, we can use the central limit theorem. If the number of samples drawn m is large, then the Monte Carlo sample mean $\bar{\theta^*}$ used to estimate E(θ) approximately follows a normal distribution with mean E(θ) and variance $\frac{Var(\theta)}{m}$. If we substitute the sample variance for Var(θ), we can get a rough estimate of our Monte Carlo standard error (or standard deviation).

Suppose we have 100 samples from our posterior distribution for θ, called θ_i^*, and that the sample variance of these draws is 5.2. A rough estimate of our Monte Carlo standard error would then be $\sqrt{\frac{5.2}{100}} \approx 0.228$. So our estimate $\bar{\theta^*}$ is probably within about 0.456 (two standard errors) of the true E(θ).

What does the standard error of our Monte Carlo estimate become if we increase our sample size to 5,000? Assume that the sample variance of the draws is still 5.2.

Report your answer to at least three decimal places.
Answer 8

```
In [51]: sample_variance <- 5.2
         sample_size <- 5000
         std_error <- sqrt(sample_variance/sample_size)
         round(std_error,3)
```

0.032

6 Next steps

After having gone through the basics of MCMC, we will go through the two most popular methods (Metropolis-Hastings algorithm, Gibbs sampler).