Chapter 12: Bayesian Modeling

Alex Chi

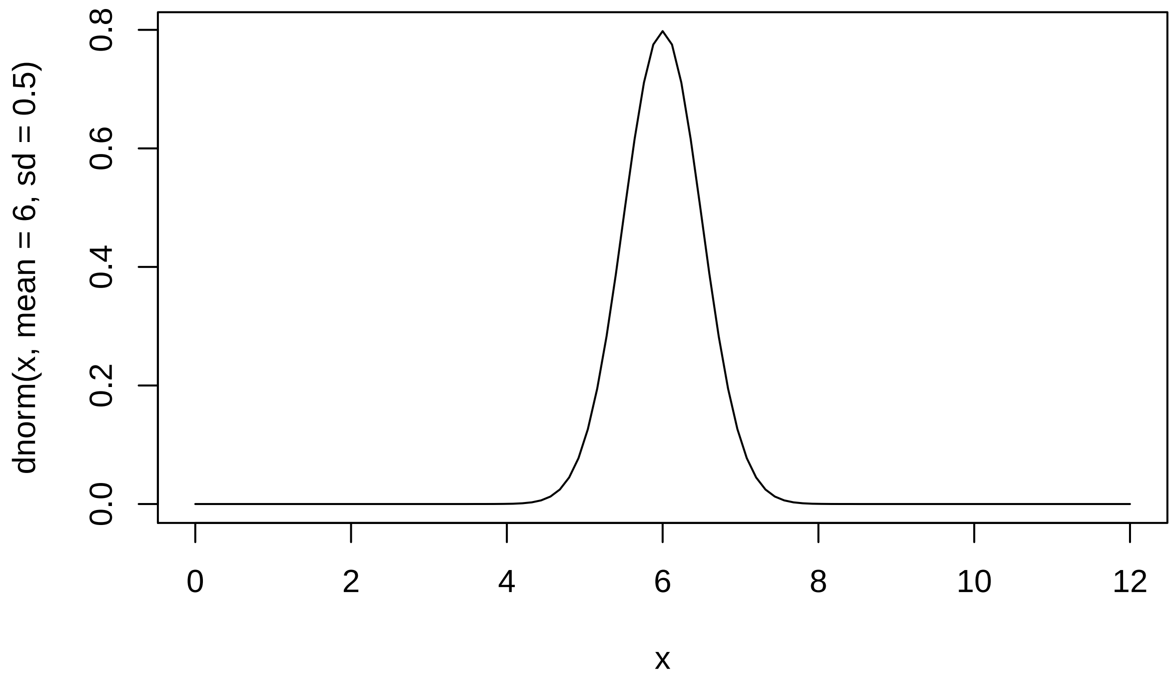
Exercises

12.1 (Learning about students sleep habits).

In Chapter 6, one is inter- ested in the mean number of hours μ slept by students at a particular college. Suppose that a particular professor's prior beliefs about μ are represented by a normal curve with mean μ0= 6 hours and variance τ2 0= 0.25.

a. Use the curve and dnorm functions to construct a plot of the prior density.

```r
curve(dnorm(x, mean=6, sd=0.5), from = 0, to = 12)
```



b. Use the qnorm function to find the quartiles of the prior density.

```r
qnorm(c(.25, .5, .75), mean=6, sd=0.5)
```

```
## [1] 5.662755 6.000000 6.337245
```

c. Find the probability (from the prior) that the mean amount of sleep ex- ceeds 7 hours (Use the pnorm function.)

```r
1 - pnorm(7, mean=6, sd=0.5)
```

```
## [1] 0.02275013
```

12.2 (Learning about students sleep habits, continued).

Suppose the number of hours slept by a sample of n students, y1,...,yn, represent a random sample from a normal density with unknown mean μ and known variance σ2.

The likelihood function of μ is given by L(μ) = exp ? n 2σ2(¯ y μ)2? .

If we combine the normal prior density with this likelihood, it can be shown that the posterior density for μ also has the normal form with updated vari- ance and mean parameters τ2 1= ?1 τ2 0 +n σ2 ?1 , μ1= τ2 1 ?μ0 τ2 0 +n¯ y σ2 ? . a. For the sleeping data collected in Chapter 6, n = 24 students were sampled and the mean sleeping time was ¯ y = 7.688. Assume that we know the sampling variance is given by σ2= 2.0. Use these values together with the prior mean and prior variance to compute the mean and variance of the posterior density of μ.

```
y = 7.688
n = 24
sigma_square = 2.0
tau_0 = 0.5
mu_0 = 6
tau_1 = sqrt((tau_0 ^ -2 + n * sigma_square ^ -1) ^ -1)
mu_1 = tau_1 ^ 2 * (mu_0 * tau_0 ^ -2 + n * y * sigma_square ^ -1)
tau_1
```

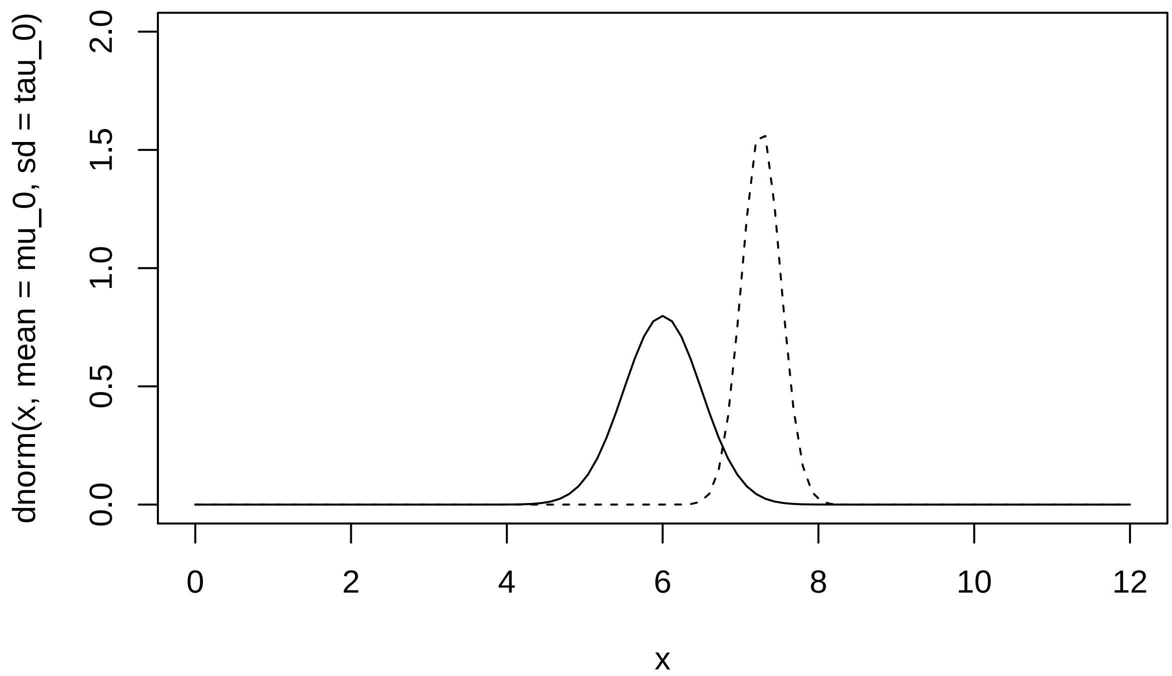
```
## [1] 0.25
```

```
mu_1
```

```
## [1] 7.266
```

b. Using two applications of the curve function, plot the prior and posterior densities together using contrasting colors.

```
curve(dnorm(x, mean=mu_0, sd=tau_0), from = 0, to = 12, ylim=c(0, 2))
curve(dnorm(x, mean=mu_1, sd=tau_1), add=TRUE, lty="dashed")
```



c. Use the qnorm function to construct a 90% posterior interval estimate of the mean sleeping time.

```
qnorm(c(0.05, 0.95), mean=mu_1, sd=tau_1)
```

```
## [1] 6.854787 7.677213
```

d. Find the posterior probability that the mean amount of sleep exceeds 7 hours.

```
1 - pnorm(7, mean=mu_1, sd=tau_1)
```

```
## [1] 0.8563357
```

12.3 (Waiting until a hit in baseball).

In sports, fans are fascinated with the patterns of streaky behavior of athletes. In baseball, a batter wishes to get a “base hit”; otherwise he records an “out.” Suppose one records the number of outs between consecutive hits (the spacing) for a particular baseball player.

For example, if the player begins the season with the outcomes H, O, O, H, H, O, O, O, O, H, O, H, H, then the spacings are given by 0, 2, 4, 1, 0 (one starts counting before the first outcome). One observes the following spacings for the player Ian Kinsler for the 2008 baseball season: 0 2 0 4 1 0 2 0 1 0 0 1 1 3 1 0 0 0 1 6 0 9 0 4 1 9 1 0 3 4 5 5 1 0 2 4 0 4 0 3 2 1 0 1 3 7 0 3 1 2 14 4 0 1 6 1 10 1 2 0 1 0 4 5 0 7 3 1 2 1 2 1 2 2 4 3 3 1 1 2 1 2 7 0 3 1 2 2 2 2 0 3 4 1 1 0 0 1 1 1 11 2 2 1 3 1 0 1 2 1 1 1 0 0 2 0 10 1 2 2 1 1 3 1 1 0 0 1 0 1 0 1 1 0 1 0 0 0 2 1 4 5 5 0 0 0 0 2 0 8 5 2 11 8 0 7 1 3 1 Let y denote the number of outs before the next base hit. A basic assumption is that y has a geometric(p) distribution with probability function f(y|p) = p(1□p)y, y = 0,1,2,… where p is the player’s hitting probability, If we assume independence in spacings, then the likelihood function of p is given by L(p) = ? y f(y|p) = pn(1□p)s, where n is the sample size and s =?y is the sum of the spacings.

a. Compute the values of n and s for Kinsler’s data.

```r
baseball.hit = c(0,2,0,4,1,0,2,0,1,0,0,1,1,3,1,0,
                 0,0,1,6,0,9,0,4,1,9,1,0,3,4,5,5,
                 1,0,2,4,0,4,0,3,2,1,0,1,3,7,0,3,
                 1,2,14,4,0,1,6,1,10,1,2,0,1,0,4,
                 5,0,7,3,1,2,1,2,1,2,2,4,3,3,1,1,
                 2,1,2,7,0,3,1,2,2,2,2,0,3,4,1,1,
                 0,0,1,1,1,11,2,2,1,3,1,0,1,2,1,
                 1,1,0,0,2,0,10,1,2,2,1,1,3,1,1,0,
                 0,1,0,1,0,1,1,0,1,0,0,0,2,1,4,5,
                 5,0,0,0,0,2,0,8,5,2,11,8,0,7,1,3,1)
n = length(baseball.hit)
s = sum(baseball.hit)
n
```

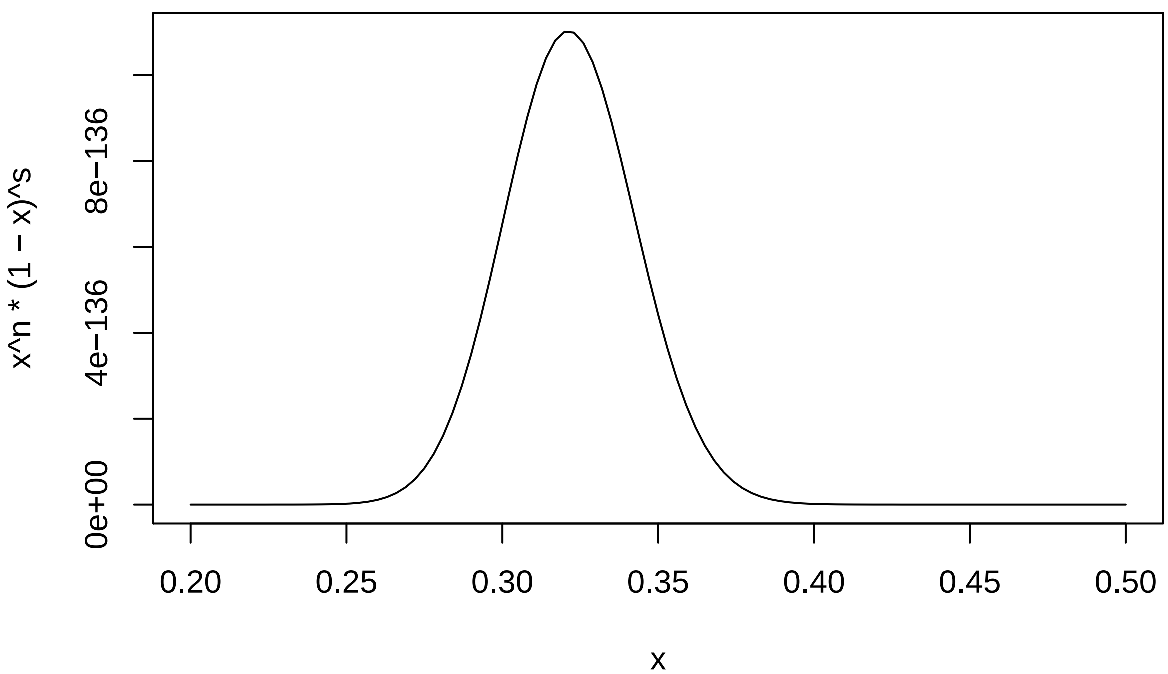
```
## [1] 159
```

```r
s
```

```
## [1] 336
```

b. Use the curve function to graph the likelihood function for values of p between 0.2 and 0.5.

```r
curve(x^n*(1-x)^s, from=0.2, to=0.5)
```



c. Based on the graph of the likelihood, which value of the hitting probability p is “most likely” given the data?

```r
optimize(function(x) { x^n*(1-x)^s }, interval=c(0.2, 0.5), maximum=TRUE)$maximum
```

```
## [1] 0.3212321
```

12.4 (Waiting until a hit, continued).

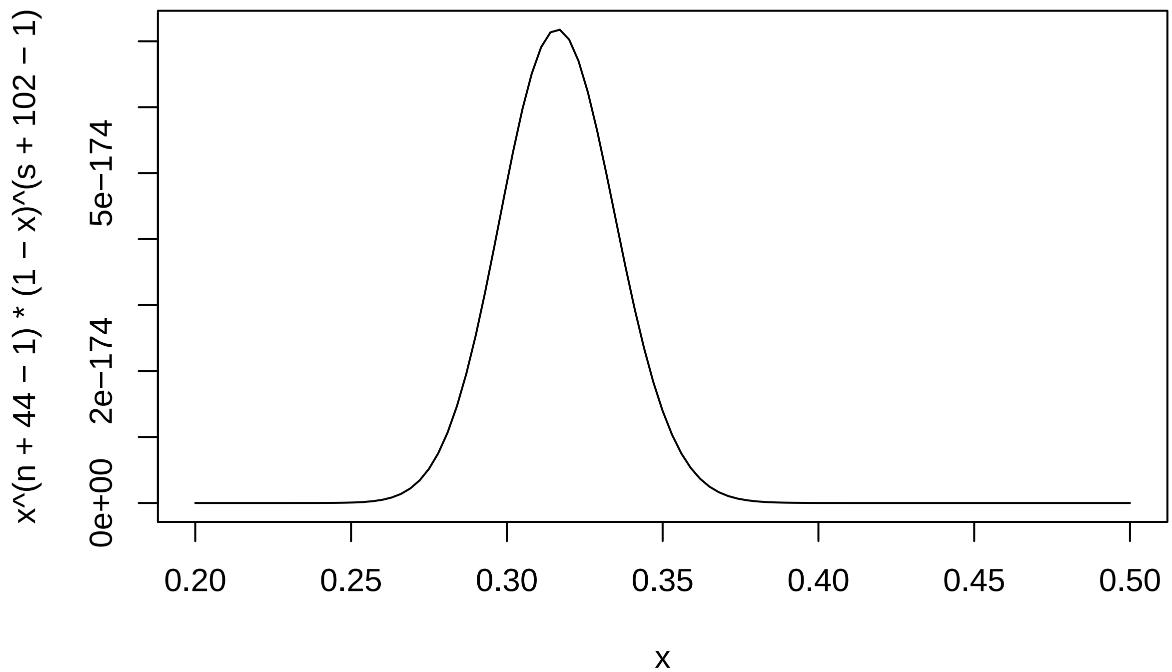
Based on Ian Kinsler's perfor- mance in previous seasons, a baseball fan has some prior beliefs about Kinsler's hitting probability p. She believes that P(p < 0.300) = 0.5 and P(p < 0.350) = 0.90. This prior information can be matched to a beta density with shape parameters a = 44 and b = 102.

g(p) = 1 B(44,102)p44□1(1□p)102□1, 0 < p < 1.

If one multiplies this prior density with the likelihood function found in Exer- cise 12.3, the posterior density for p is given (up to a proportionality constant) by g(p|data) □ pn+44□1(1□p)s+102□1, 0 < p < 1, where n is the sample size and s =?y is the sum of the spacings. This is a beta density with shape parameters a = n + 44 and b = s + 102. (The values of n and s are found from the data in Exercise 12.3.)

a. Using the curve function, graph the posterior density for values of the hitting probability p between values 0.2 and 0.5.

```
curve(x^(n+44-1)*(1-x)^(s+102-1), from=0.2, to=0.5)
```



b. Using the qbeta function, find the median of the posterior density of p.

```
qbeta(0.5,n+44, s+102)
```

```
## [1] 0.3165019
```

c. Using the qbeta function, construct a 95% Bayesian interval estimate for p

```
qbeta(c(0.025, 1-0.025),n+44, s+102)
```

```
## [1] 0.2812654 0.3532033
```

12.5 (Waiting until a hit, continued).

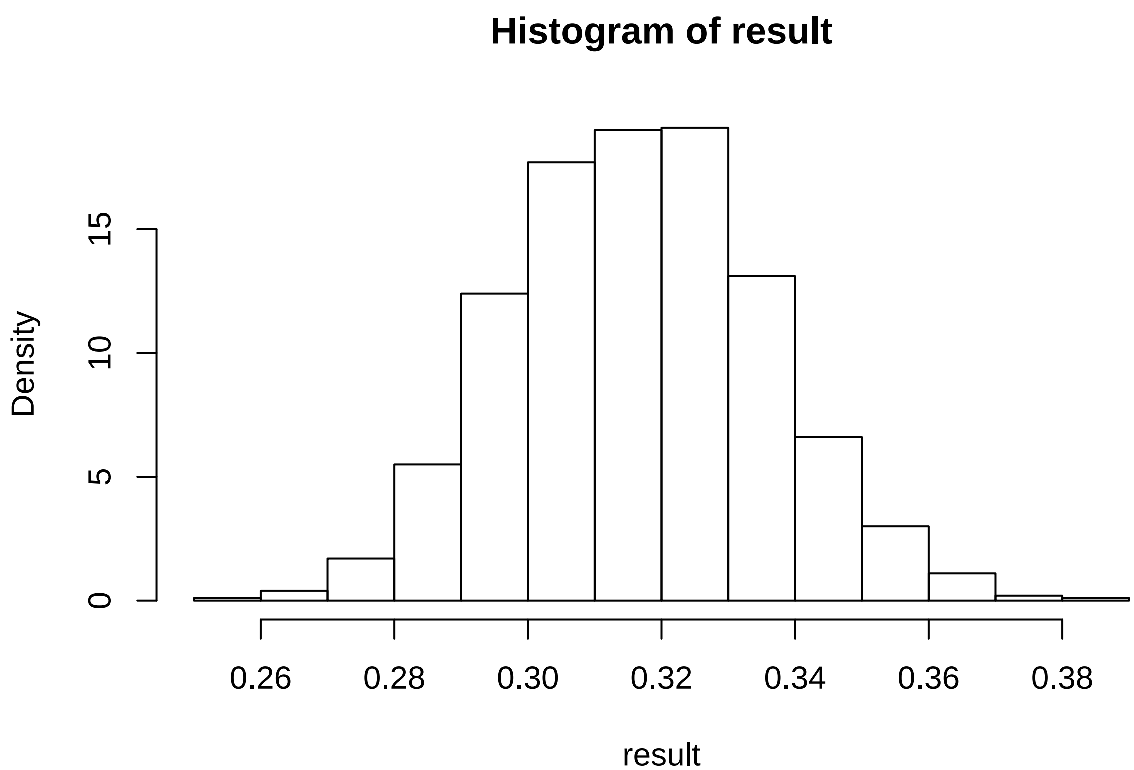
In Exercise 12.4, we saw that the posterior density for Ian Kinsler's hitting probability is a beta density with shape parameters a = n+44 and b = s+102. (The values of n and s are found from the data in Exercise 12.3.)

a. Using the rbeta function, simulate 1000 values from the posterior density of p.

```
result = rbeta(1000, n+44, s+102)
```

b. Use the hist function on the simulated sample to display the posterior density.

```
hist(result, freq=FALSE)
```



c. Using the simulated draws, approximate the mean and standard deviation of the posterior density.

```
mean(result)
```

```
## [1] 0.3166756
```

```
sd(result)
```

```
## [1] 0.01899016
```

d. Using the simulated draws, construct a 95% Bayesian interval estimate. Compare the interval with the exact 95% interval estimate using the qbeta function.

```
quantile(result, c(0.025, 1-0.025))
```

```
##      2.5%     97.5%
## 0.2810376 0.3532469
```

```r
qbeta(c(0.025, 1-0.025), n+44, s+102)
```

```
## [1] 0.2812654 0.3532033
```

12.6 (Waiting until a hit, continued).

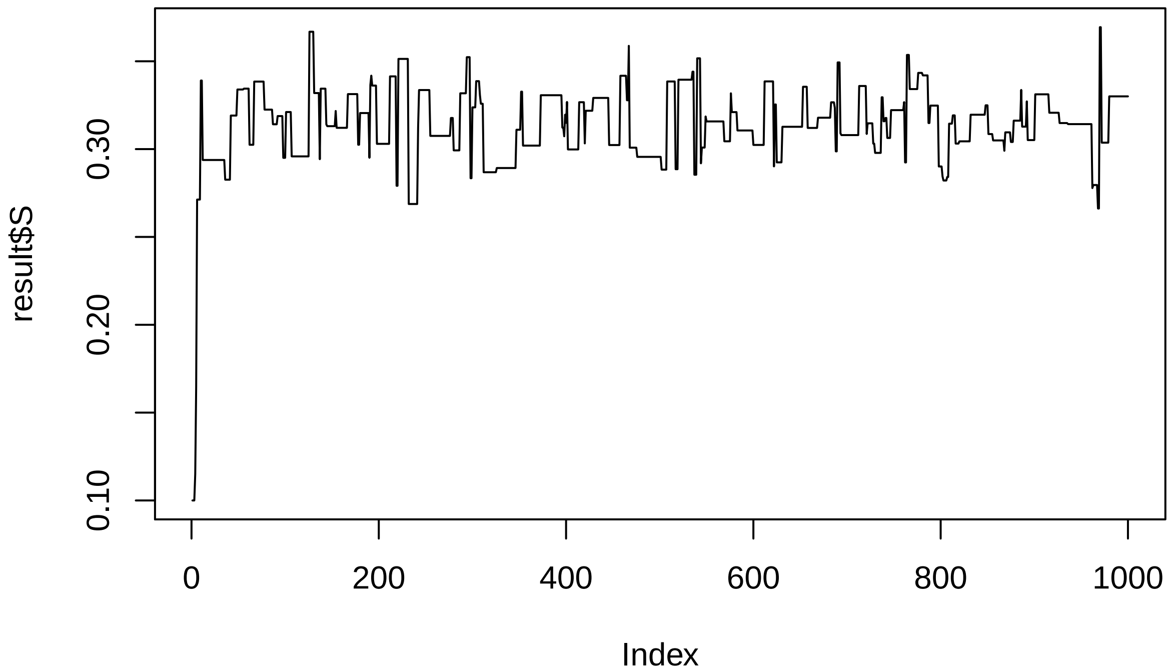
In Exercise 12.4, we saw that the posterior density for Ian Kinsler's hitting probability is a beta density with shape parameters a = n+44 and b = s+102. (The values of n and s are found from the data in Exercise 12.3.) The function metrop.hasting.rw described in the chapter can be used to simulate a sample from the posterior density of the hitting probability p.

The following function betalogpost will compute the logarithm of the beta density with shape parameters a and b.

a. Use the function metrop.hasting.rw together with the function betalog- post to simulate from the posterior density using the Metropolis Hastings random walk algorithm. Use p = 0.2 as a starting value, take 1000 itera- tions, and use the scale constant C = 0.1. Construct a trace plot of the simulated values and find the acceptance rate of the algorithm. Compute the posterior mean of p from the simulated draws and compare the simu- lation estimate with the exact posterior mean (n+44)/(n+s+44+102).

```r
metrop.hasting.rw = function(logpost, current, C, iter, ...){
  S = rep(0, iter); n.accept = 0
  for(j in 1:iter) {
    candidate = runif(1, min=current - C, max=current + C)
    prob = exp(logpost(candidate, ...) - logpost(current, ...))
    accept = ifelse(runif(1) < prob, "yes", "no")
    current = ifelse(accept == "yes", candidate, current)
    S[j] = current; n.accept = n.accept + (accept == "yes")
  }
  list(S=S, accept.rate=n.accept / iter)
}
betalogpost = function(p, a, b) dbeta(p, a, b, log=TRUE)

result = metrop.hasting.rw(betalogpost, 0.1, 0.2, 1000, n+44, s+102)
plot(result$S, type="l")
```




```r
mean(result$S)
```

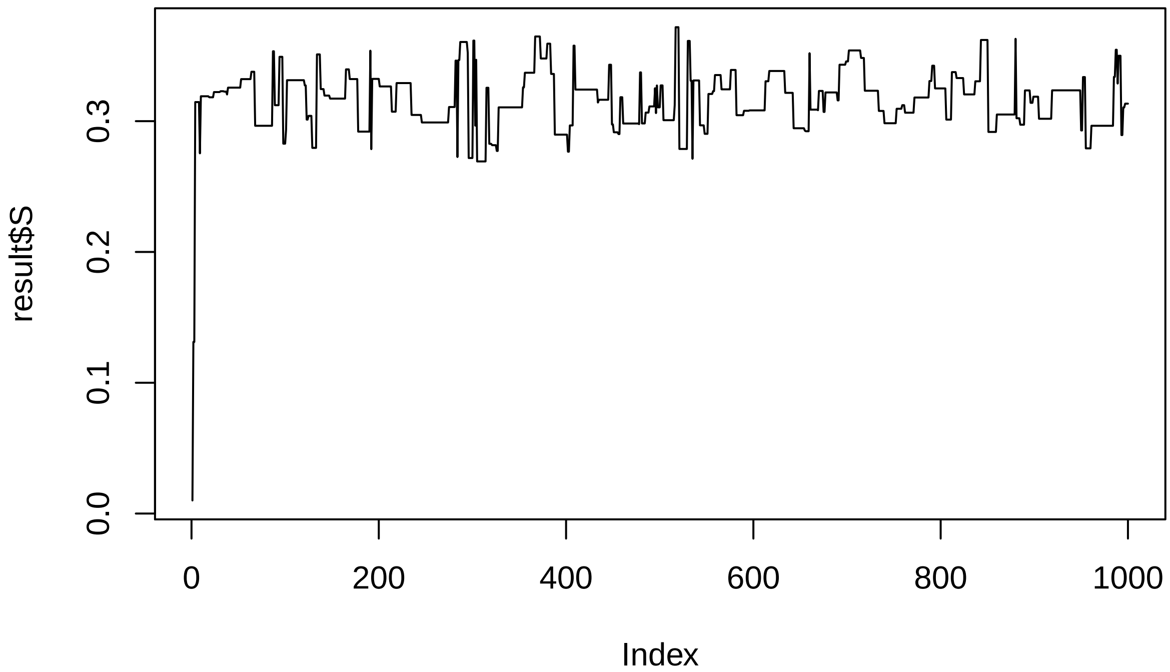
```
## [1] 0.3138593
```

```r
(n+44)/(n+s+44+102)
```

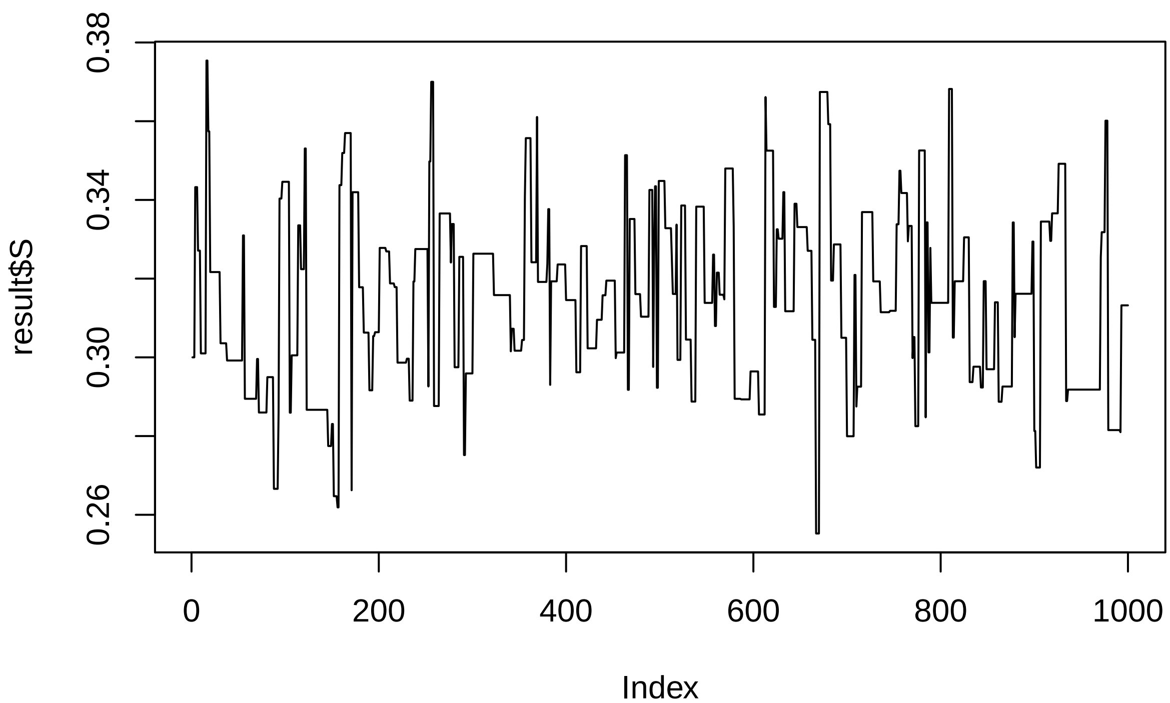
```
## [1] 0.3166927
```

b. Rerun the random walk algorithm using the alternative scale constant values C = 0.01 and C = 0.30. In each case, construct a trace plot and compute the acceptance rate of the algorithm. Of the three choices for the scale constant C, are any of the values unsuitable? Explain.

```r
C010 = result$accept.rate
result = metrop.hasting.rw(betalogpost, 0.01, 0.2, 1000, n+44, s+102)
plot(result$S, type="l")
```




```r
C001 = result$accept.rate
result = metrop.hasting.rw(betalogpost, 0.30, 0.2, 1000, n+44, s+102)
plot(result$S, type="l")
```




```r
C030 = result$accept.rate
c(C010, C001, C030)
```

```
## [1] 0.149 0.155 0.183
```