The Metropolis algorithm

Stat 340: Bayesian Statistics

Big idea: Markov chain simulation

Situation: Suppose that sampling from $\pi_n(\theta)$ is hard, but that we can (somehow) generate a Markov chain $\{\theta(t), t \in T\}$ with stationary distribution $\pi_n(\theta)$.

- We know the the stationary distribution
- We seek transitions $p(\theta^{(t+1)}|\theta^{(t)})$ that will take us to the stationary distribution

Overview

- Start from some initial guess $\theta^{(0)}$ and let the chain run for n steps (n large), so that it reaches its stationary distribution
- After convergence, all additional steps in the chain are draws from the stationary distribution $\pi_n(\theta)$
- MCMC methods are all based on the this idea; difference is just in how the transitions in the MC are created

Example: Launch failures

- FAA and USAF were interested in estimating the failure probability for new rockets launched by companies with limited experience
- Goal is to assess prelaunch risk.
- Failures have significant impact on
  - public safety
  - aerospace manufacturer's ability to develop and field new rocket systems.
- Johnson et al. (2005) data from 1980-2002
  - 11 launches: 3 successes, 8 failures

Model

$Y = \#$ successful launches

Likelihood: Assuming trials are iid $\text{Bernoulli}(\theta)$

$$Y \sim \text{Binomial}(n = 11, \theta)$$

Prior: Elicitation leads to uniform on (0.1, 0.9)

Posterior:

$$p(\theta|y) \propto \begin{cases} \theta^3(1-\theta)^8 & \text{if } 0.1 < \theta < 0.9 \\ 0 & \text{otherwise.} \end{cases}$$

Is the posterior a density we know?

Metropolis algorithm

1. Select a value $\theta^{(0)}$ where $\pi_n(\theta^{(0)}) > 0$

2. Given the current draw $\theta^{(i)}$, propose a *candidate draw* $\theta^p \sim \text{Unif}(\theta^{(i)} + C, \theta^{(i)} + C)$.

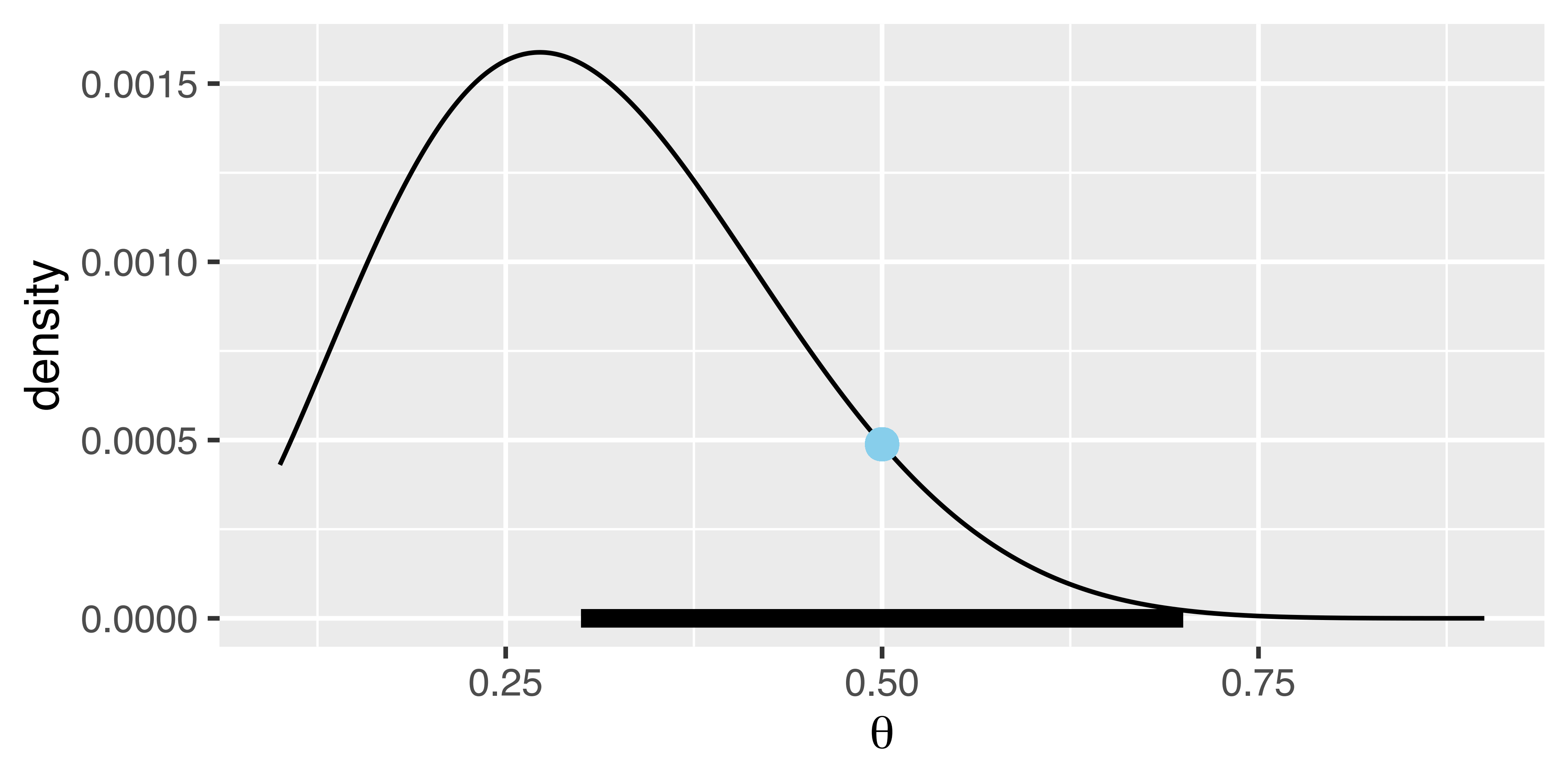
3. Evaluate the (unnormalized) posterior at the current value: $\pi_n(\theta^{(i)})$.

4. Evaluate the (unnormalized) posterior at the candidate: $\pi_n(\theta^c)$.

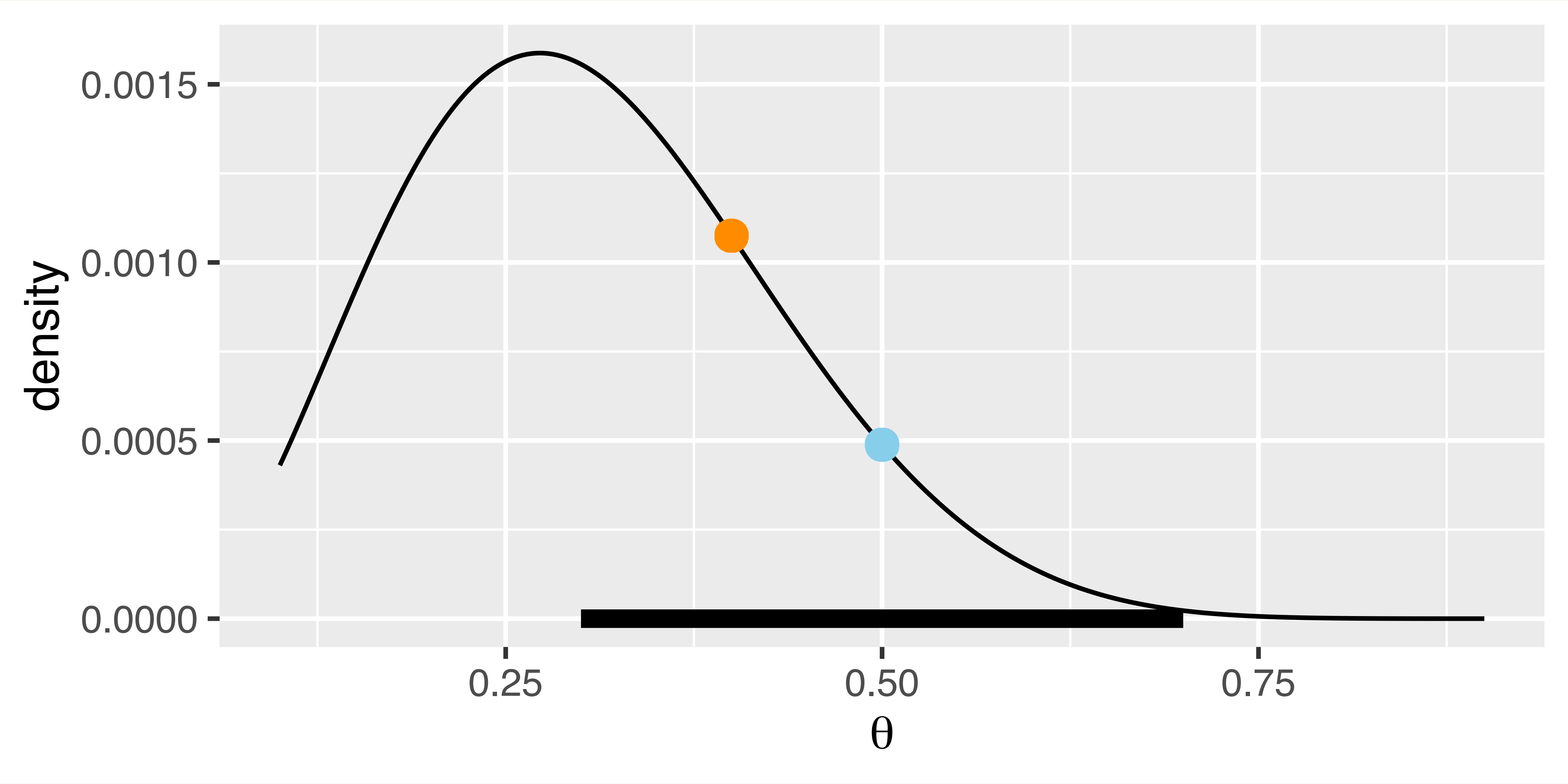
5. Accept candidate with probability $R = \min\left\{\pi_n(\theta^p)/\pi_n(\theta^{(i)}), 1\right\}$.

    - Draw $U \sim \text{Unif}(0, 1)$, if $U < R$ set $\theta^{(i+1)} = \theta^p$
    - Otherwise, set $\theta^{(i+1)} = \theta^{(i)}$.

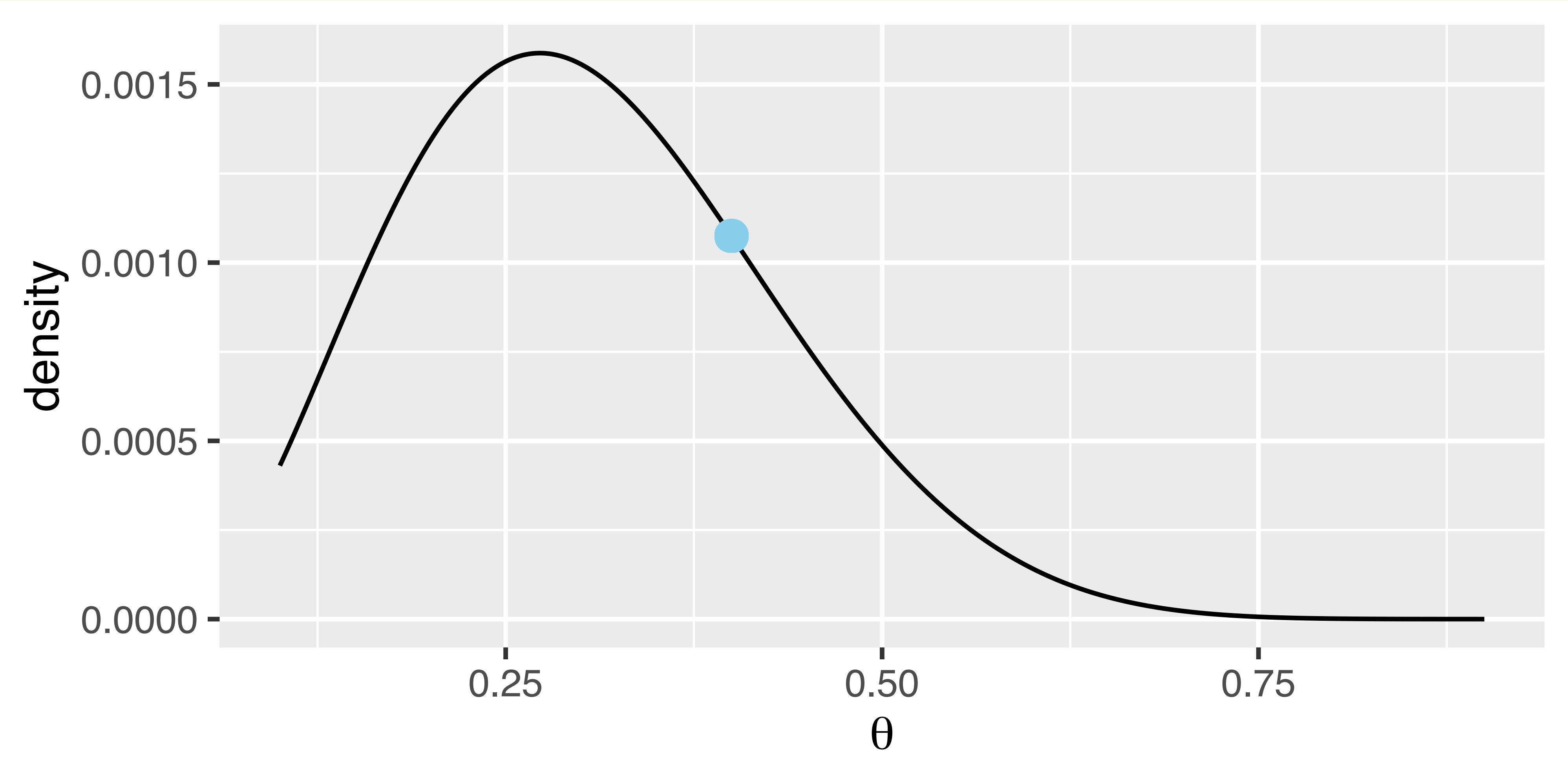
Initial value: 0.5



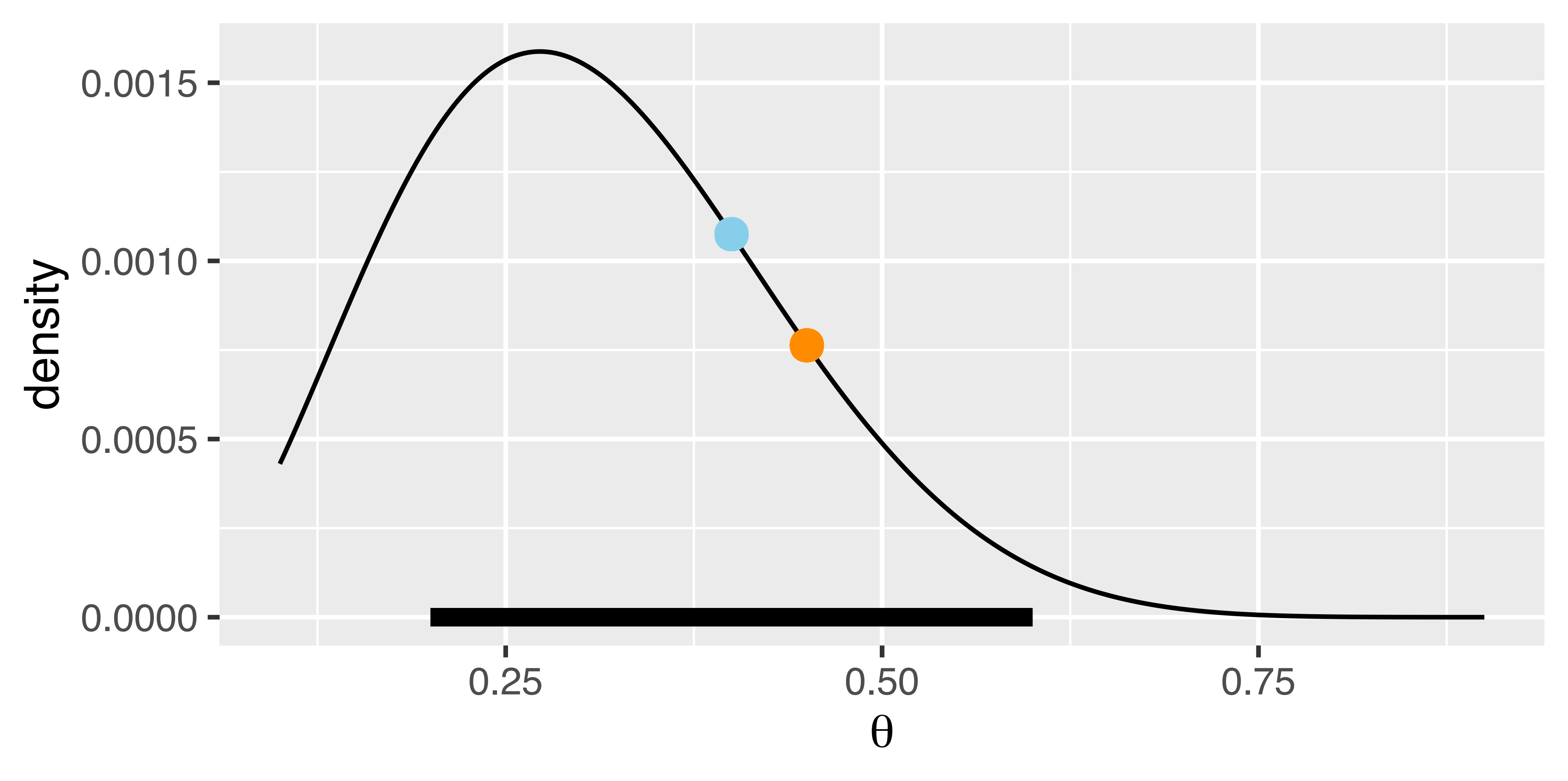
Propose 0.4, acceptance probability = 1



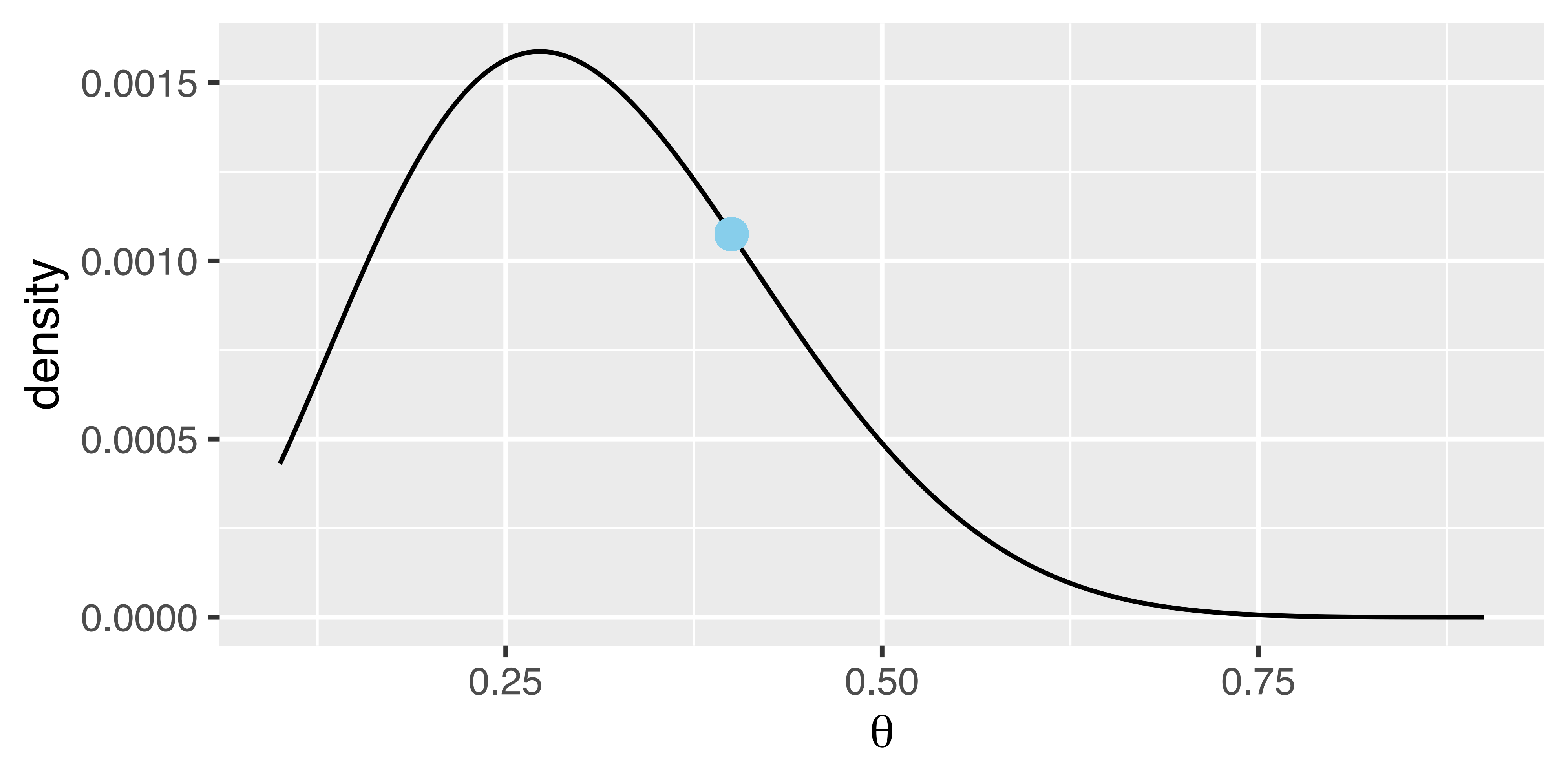
Update current draw



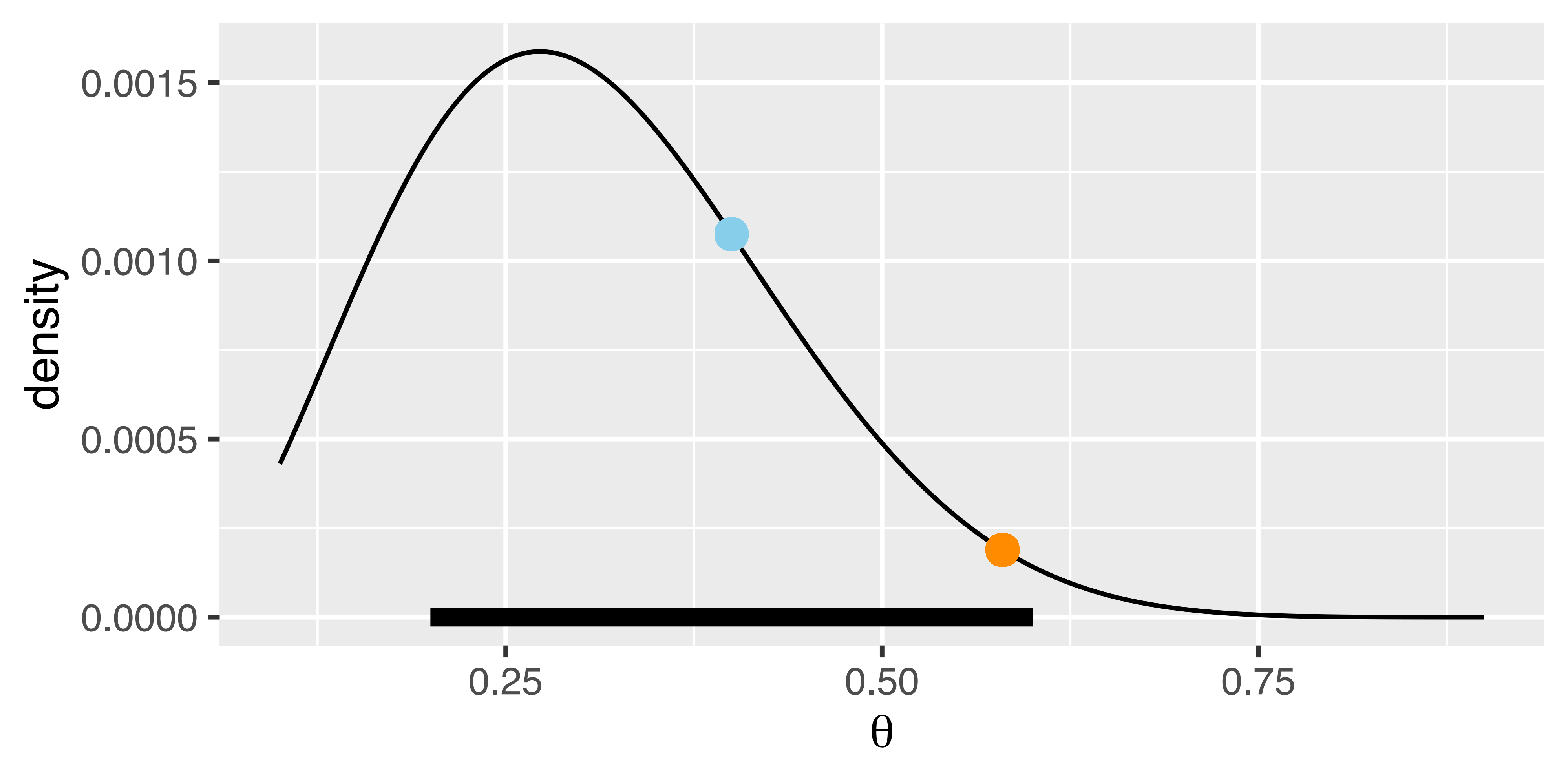
Propose 0.45, acceptance probability 0.71



$U = 0.8$, retain current draw



Propose 0.58, acceptance probability 0.18



Metropolis function (Albert and Hu, p. 326)

```r
metropolis <- function(logpost, current, C, iter, ...){
  S <- rep(0, iter) # container for draws
  n_accept <- 0     # acceptance counter

  # Iterate through candidate draws
  for(j in 1:iter){
  candidate <- runif(1, min = current - C, max = current + C)
  prob <- exp(logpost(candidate, ...) -
             logpost(current, ...))

  if(is.nan(prob)) prob <- 0 # deal with draws outside parameter space

  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) # Return draws and acceptance rate
}
```

Using `metropolis()`

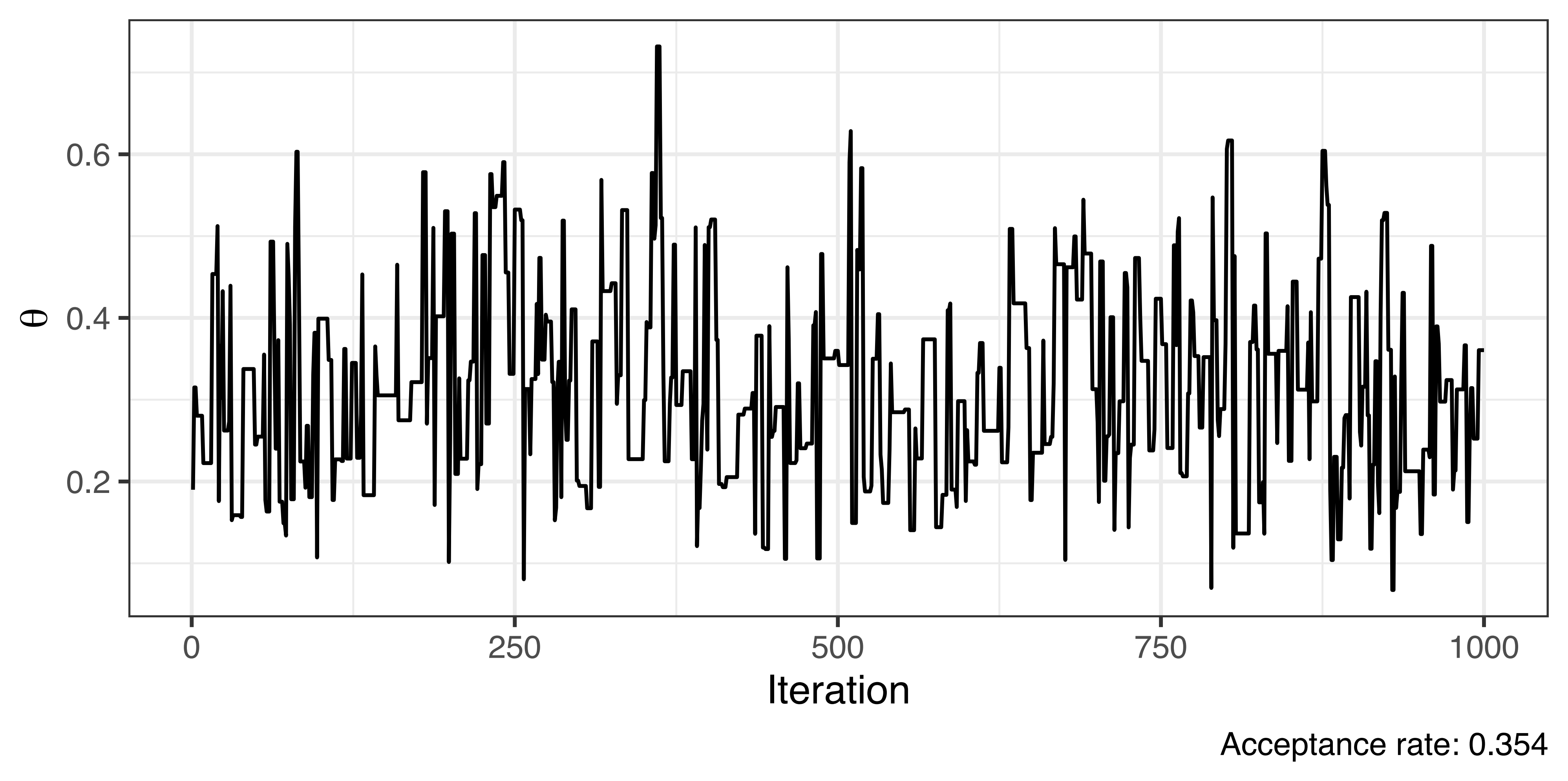
Writing the log-posterior function

```
# Log posterior function
log_posterior <- function(.theta, samp) {
  dbinom(samp$y, size = samp$n, prob = .theta, log = TRUE) + dunif(.theta, 0.1, 0.9)
}
```

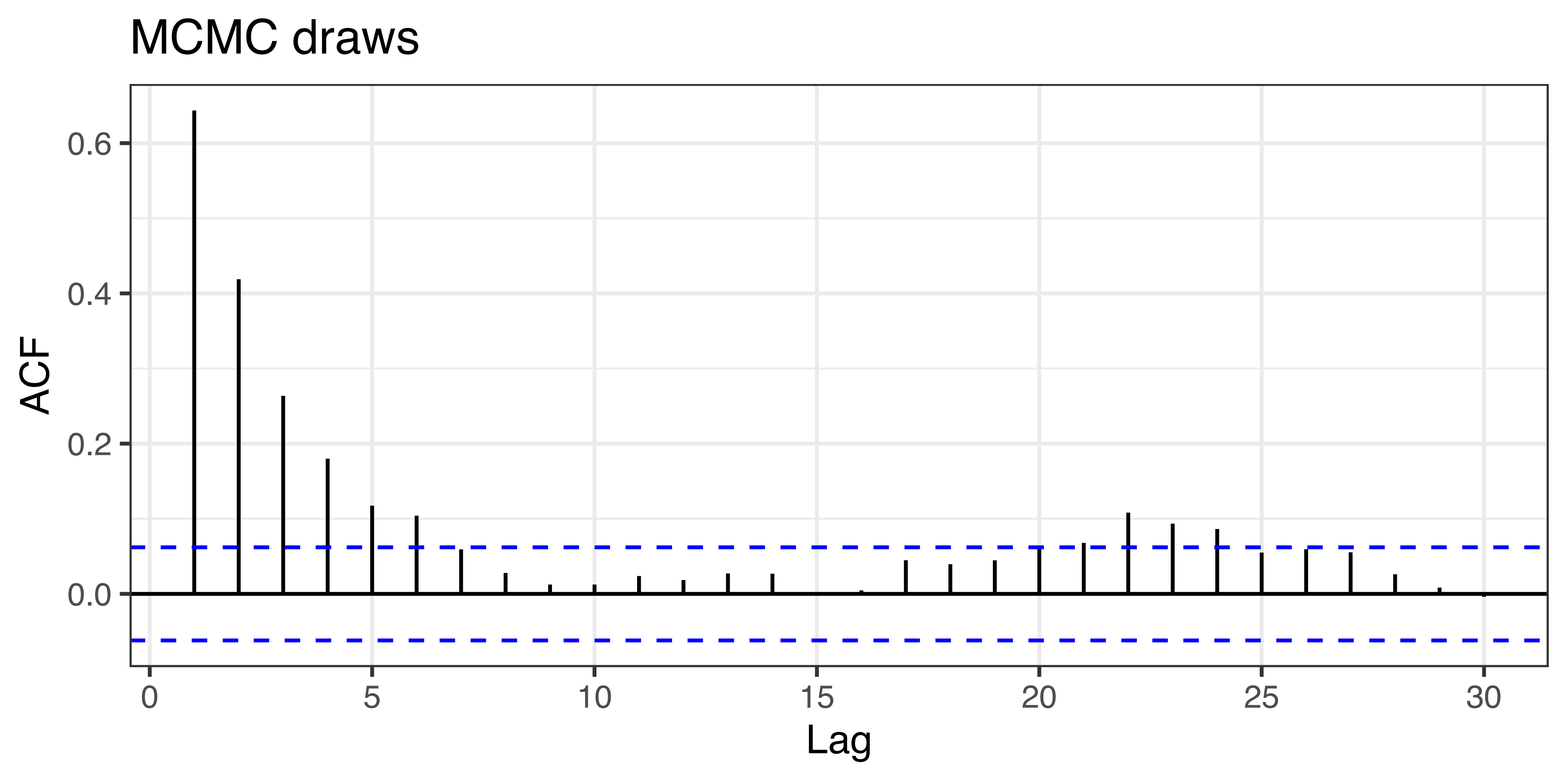
Next, initialize `current`, `C`, `iter`, and pass in the necessary data as the last argument to `metropolis`:

```
# Running the sampler
set.seed(57948)                          # for reproducibility
samp_stats <- list(y = 3, n = 11)  # sample data
mcmc_draws <- metropolis(logpost = log_posterior, current = 0.5, C = 0.5,
                         iter = 1000, samp_stats)
```

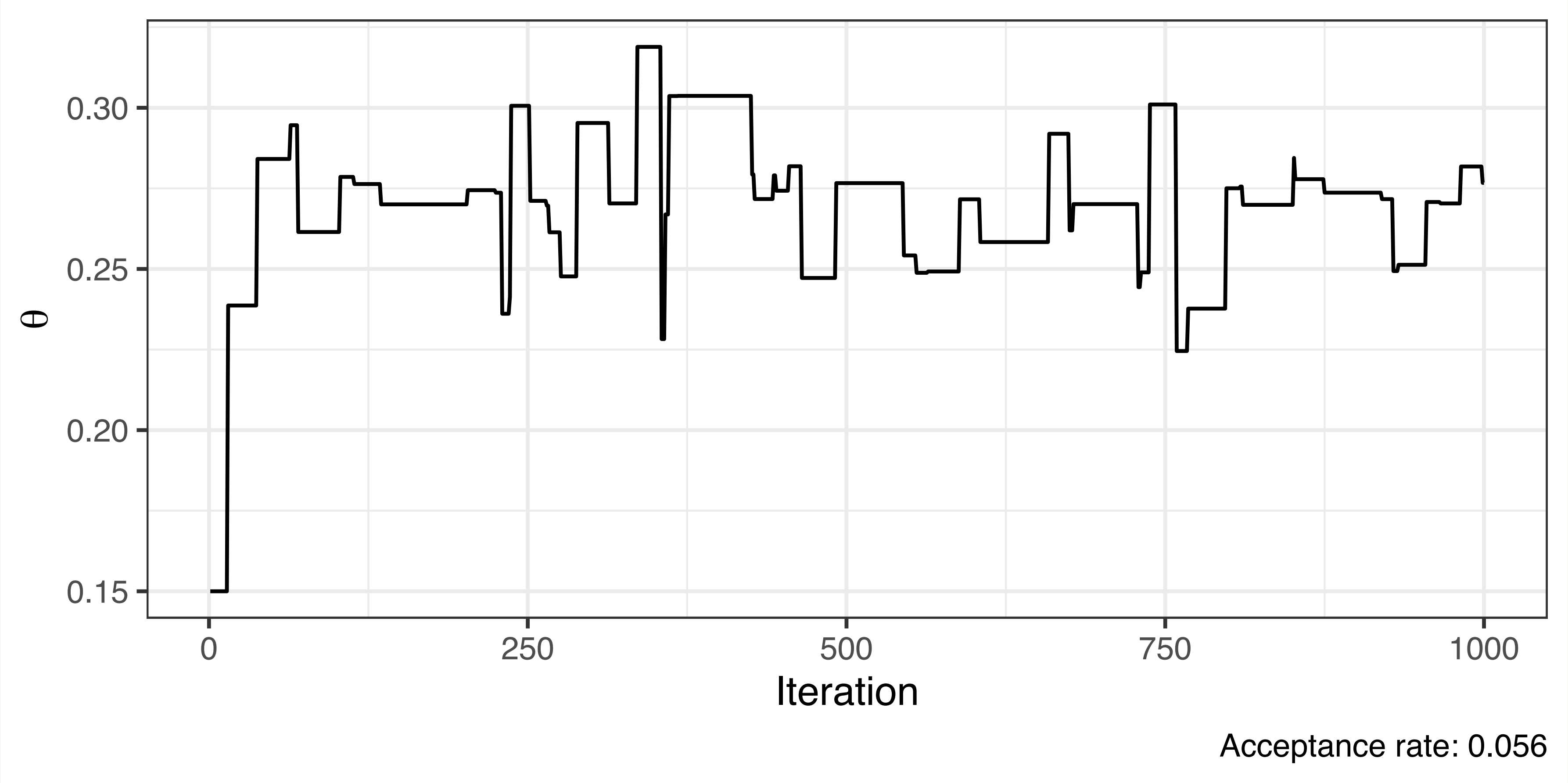
Did the sampler work?



Did the sampler work?

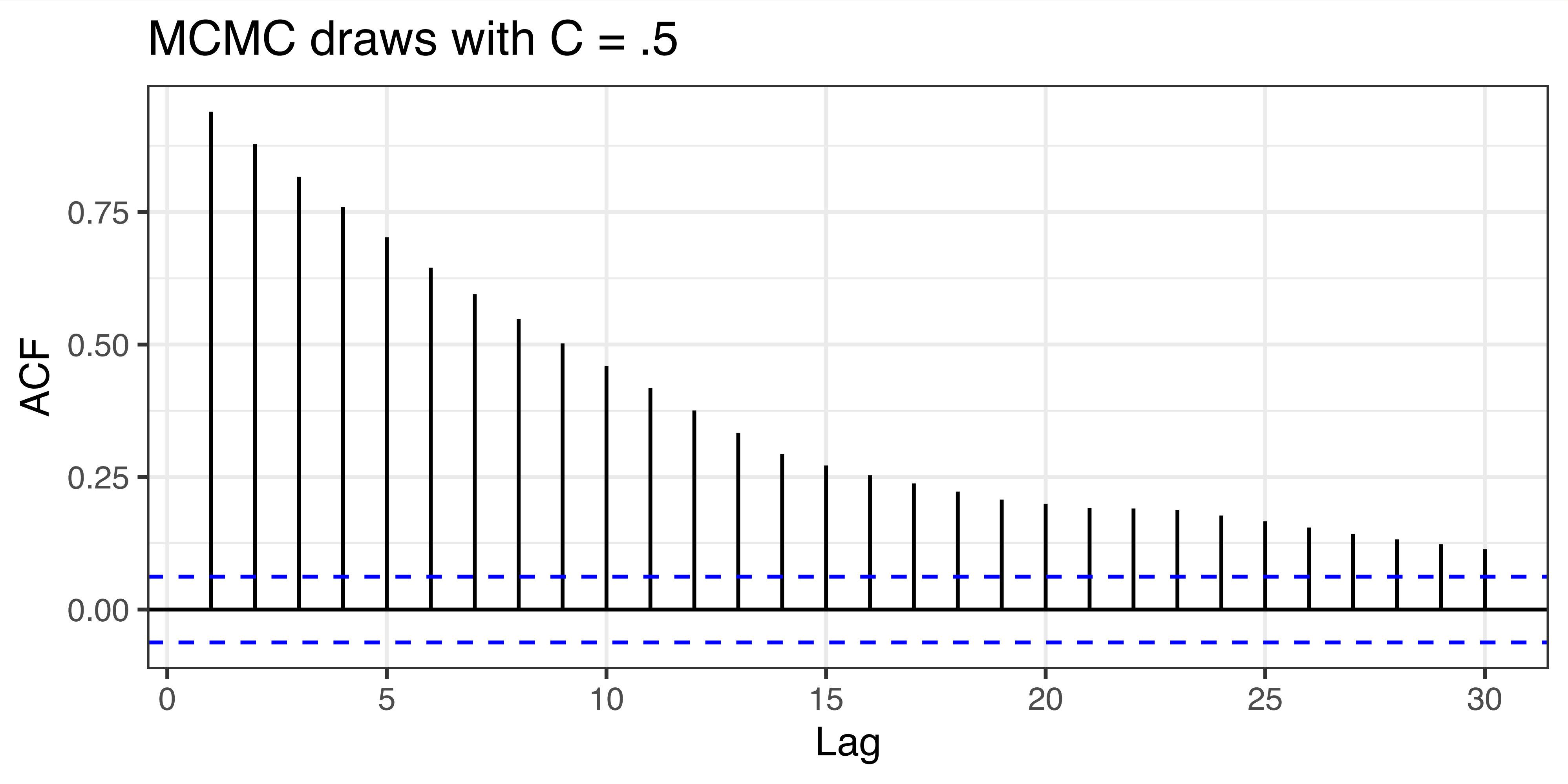


Now suppose we observed 300 successes and 800 failures and ran our Metropolis sampler (`current = 0.15`, `C = 0.5`)

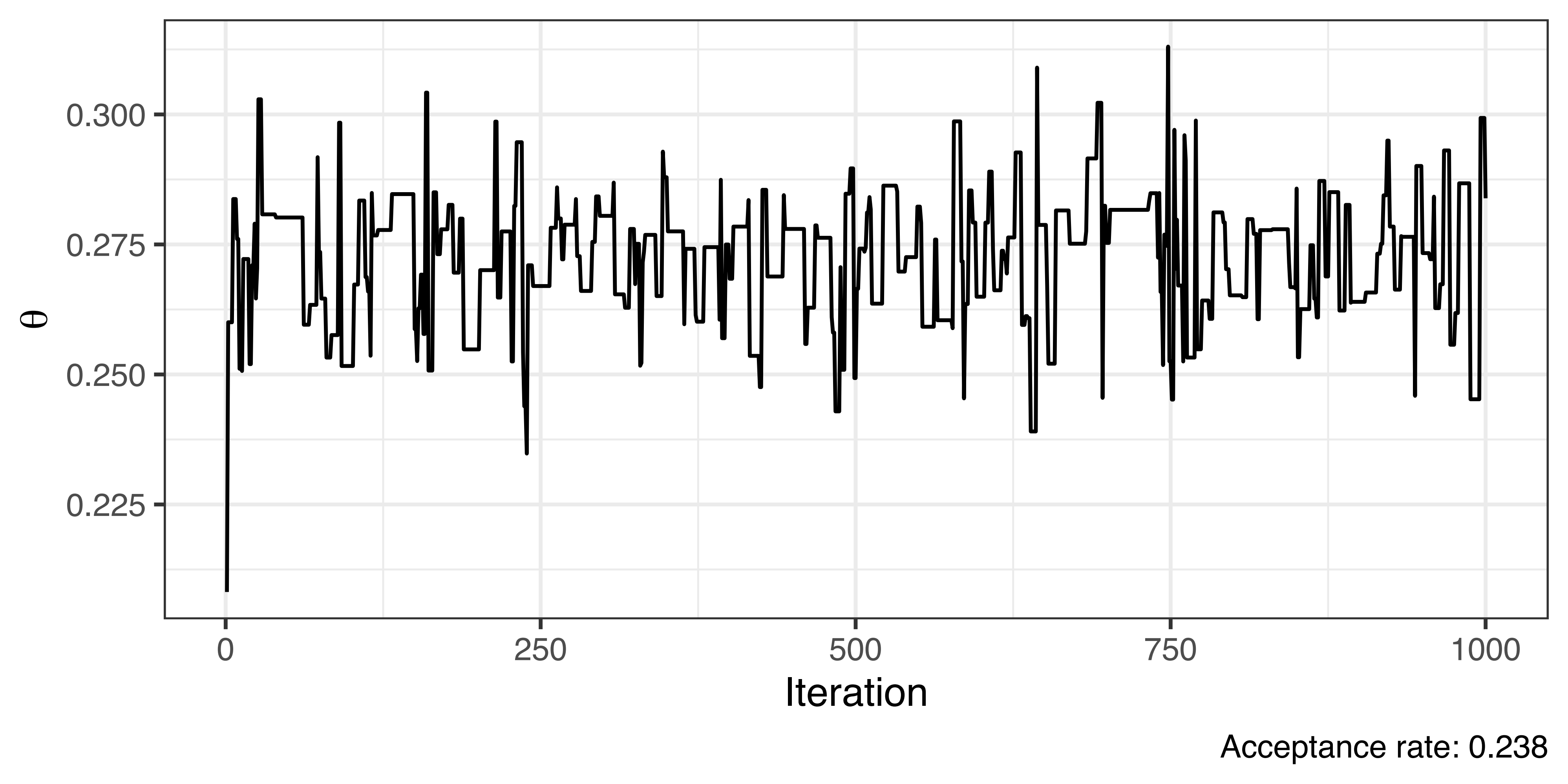


Are you comfortable with this chain?

How does the ACF plot look?

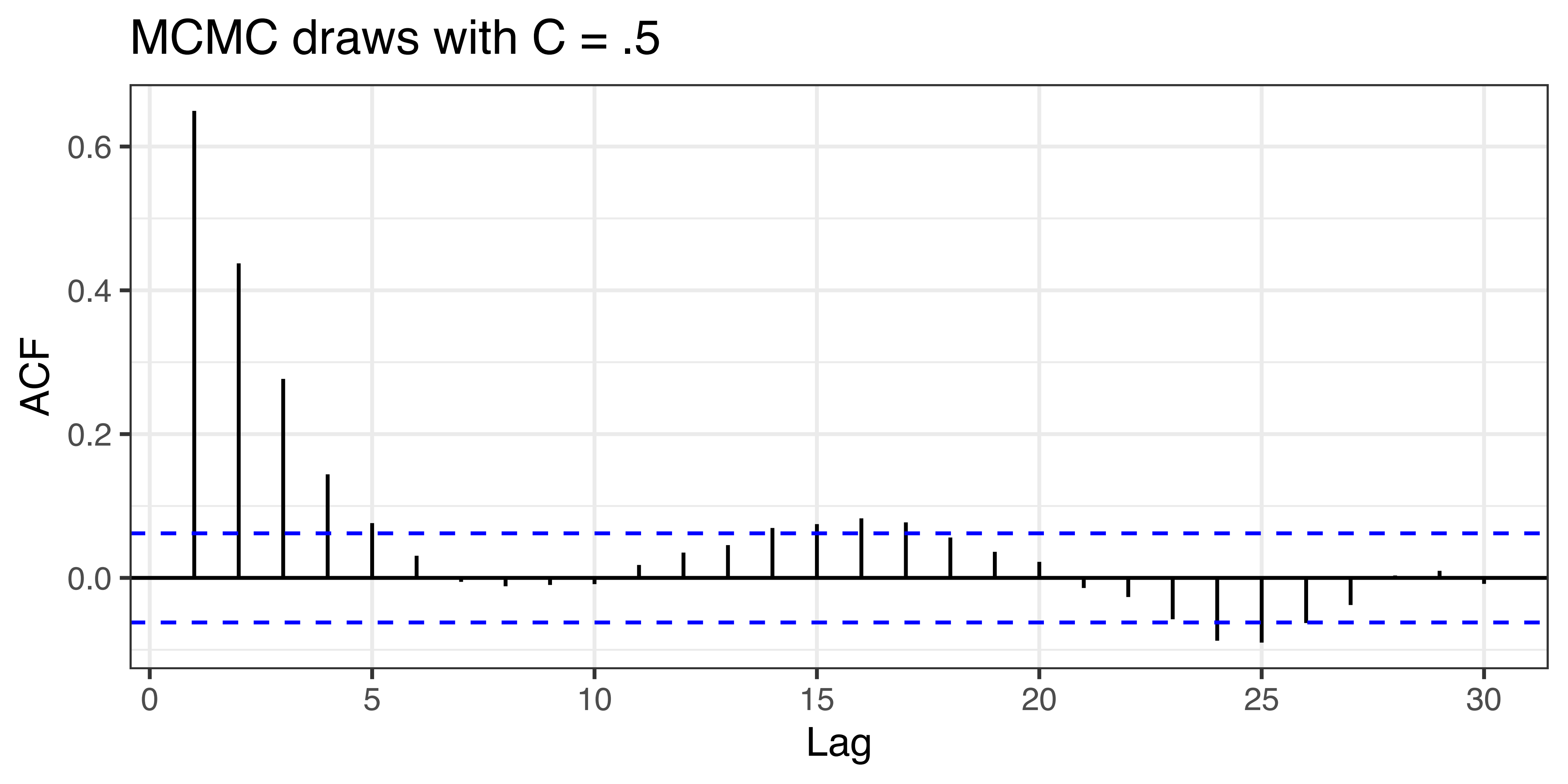


Setting `C = .1`

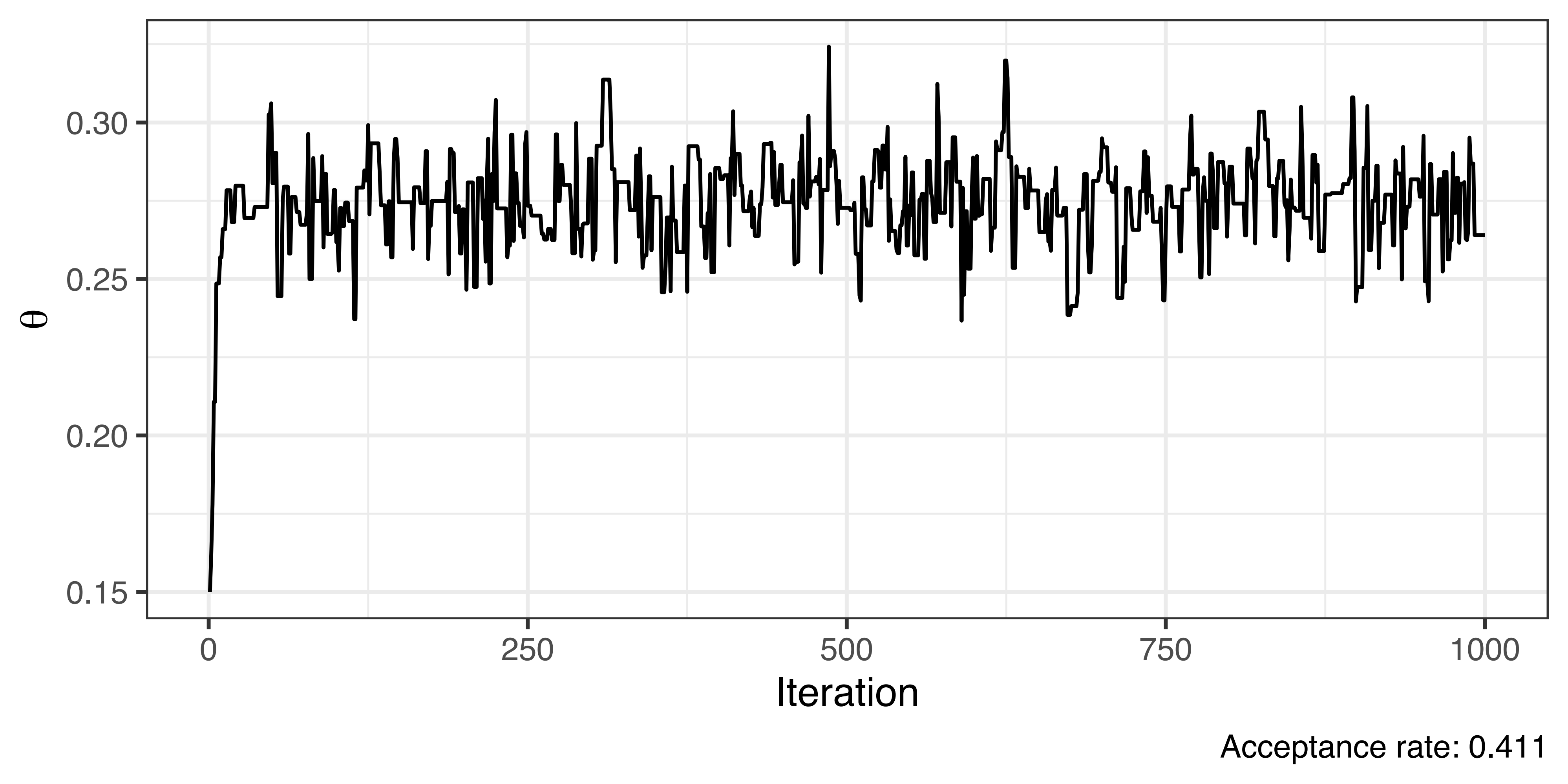


0.300
0.275
0.250
0.225
θ
0
250
500
750
1000
Iteration
Acceptance rate: 0.238

Setting `C = .1`

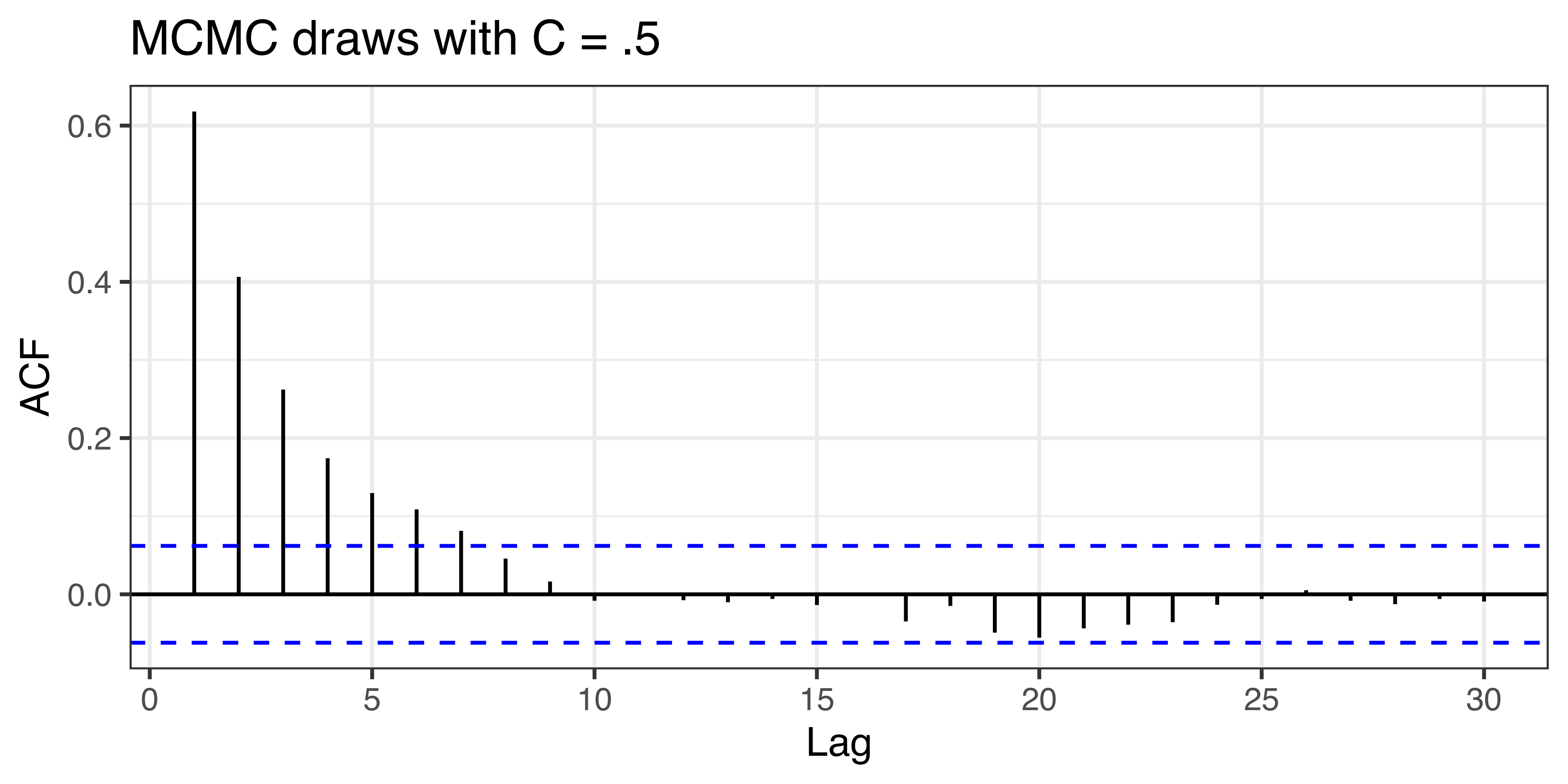


Setting `C = .05`

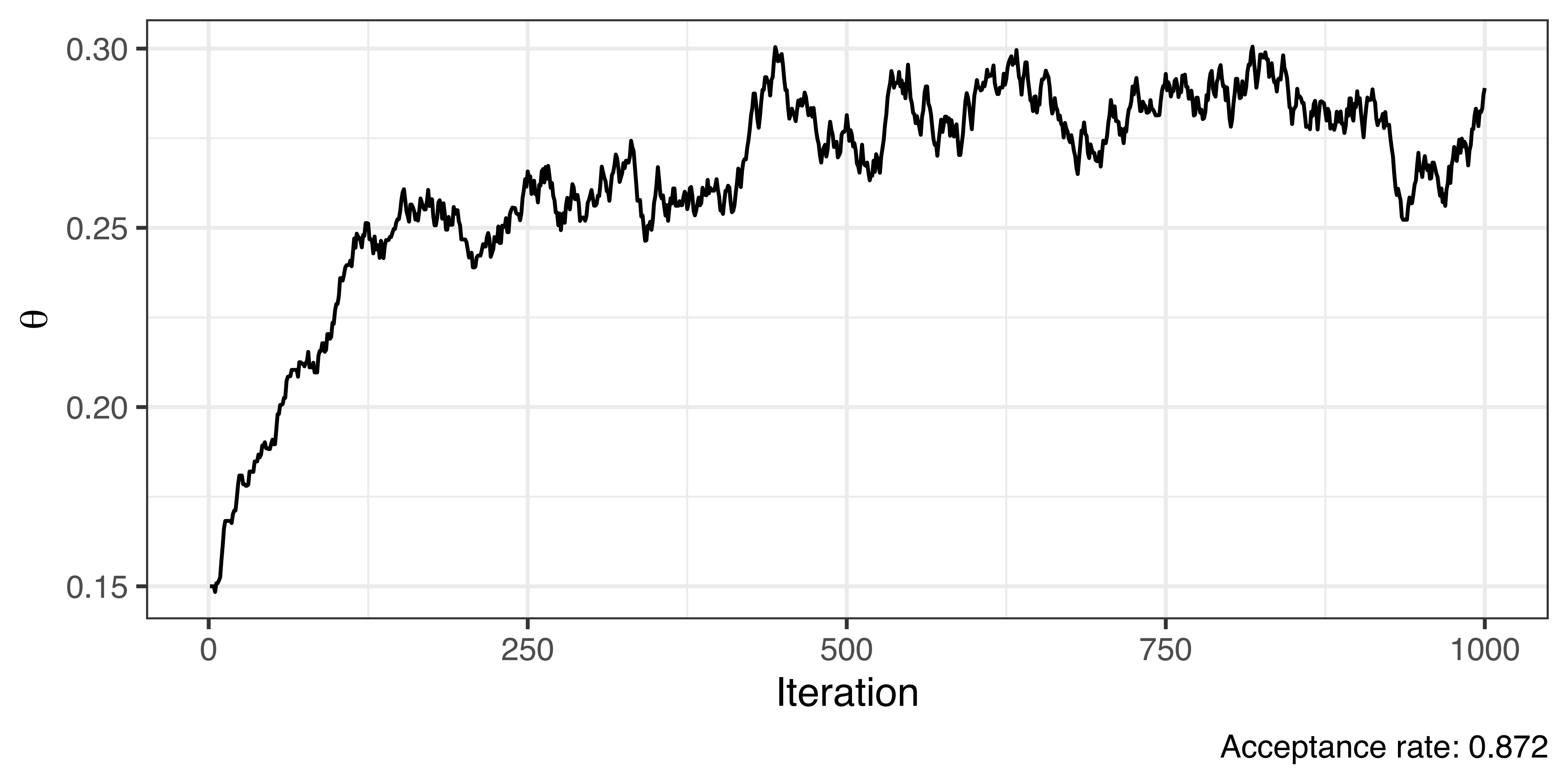


0.30
0.25
0.20
0.15
θ
0
250
500
750
1000
Iteration
Acceptance rate: 0.411

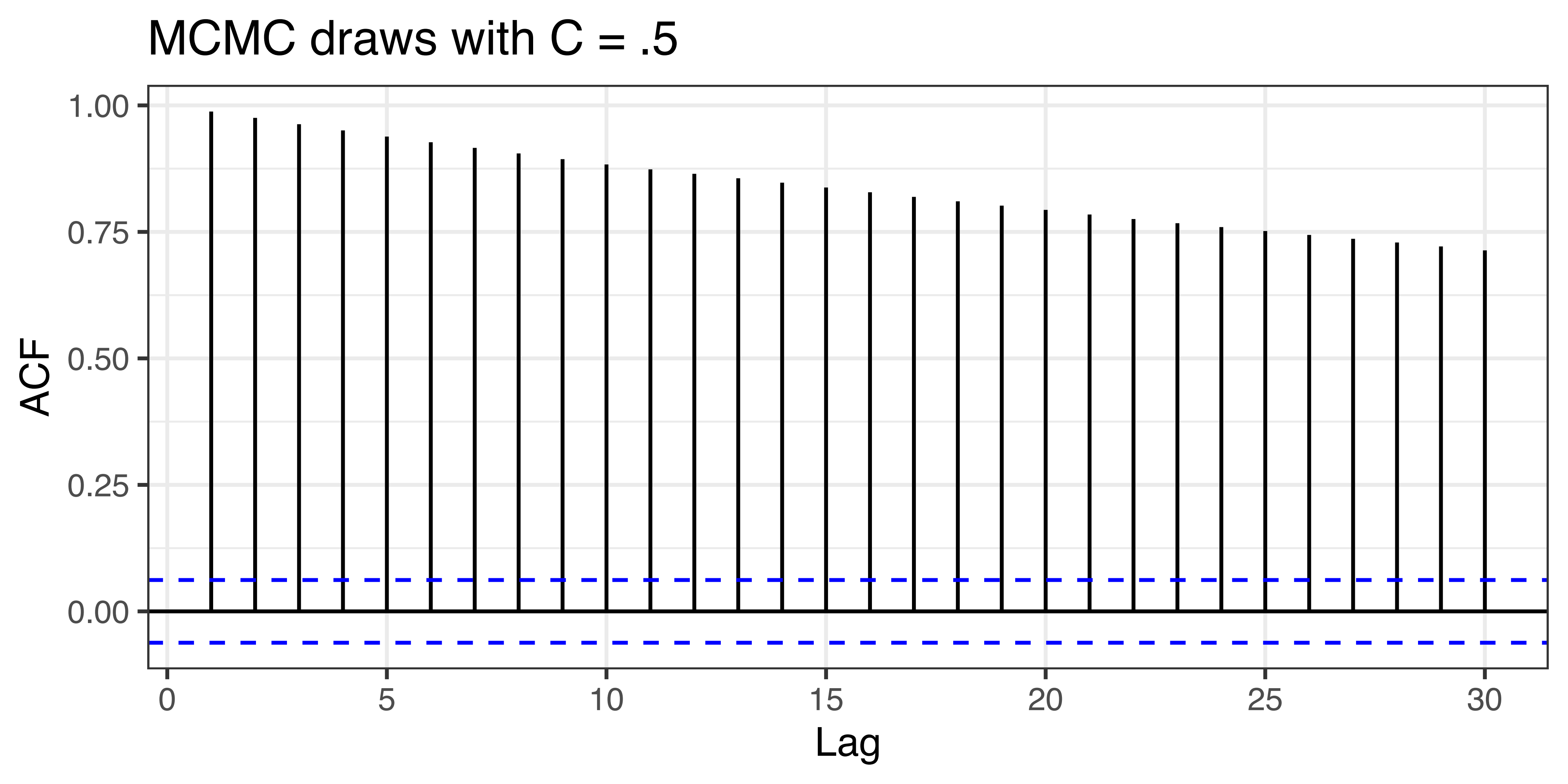
Setting `C = .05`



Setting `C = .005`



Setting `C = .005`



If the sampler worked...

Conduct inference just like when we had draws from the grid approximate posterior

Toss out burn-in period first!

Credible intervals

```r
quantile(mcmc_draws$S[-c(1:100)],
         probs = c(0.05, 0.95))
```

```
##        5%       95%
## 0.1437817 0.5318214
```

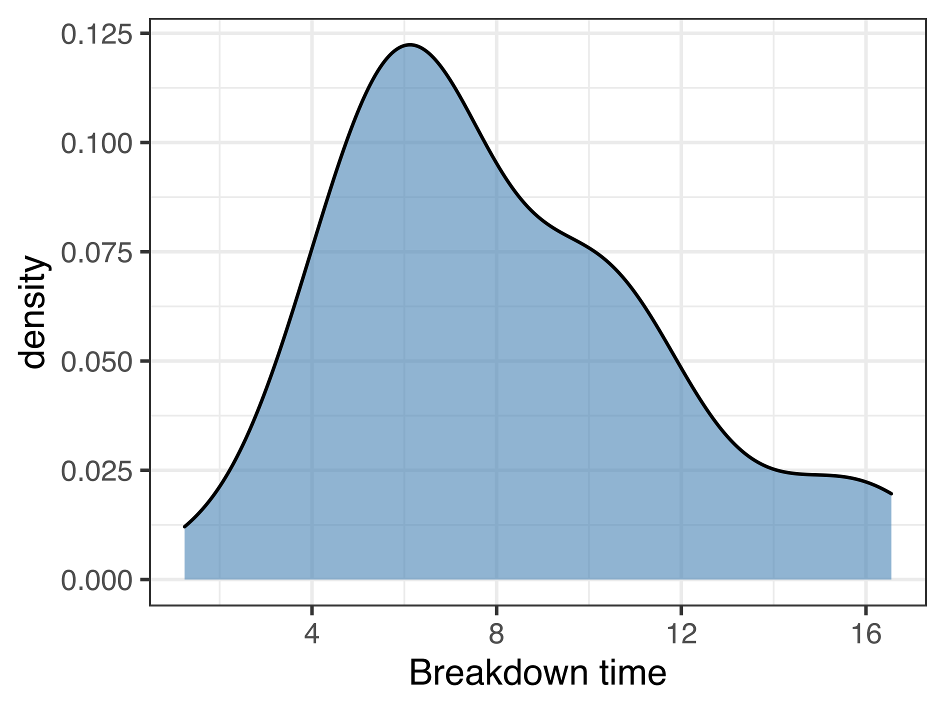
Posterior probabilities

```r
mean(mcmc_draws$S[-c(1:100)] > 0.5)
```

```
## [1] 0.09333333
```

Example: Fluid breakdown

- Engineers needed to understand how long machines can run before replacing oil in a factory
- Collected viscosity breakdown times (in thousands of hours) for 50 samples



Model

Let T_i denote the breakdown time (thousands of hours) and $Y_i = \log(T_i)$

Likelihood:

$$T_i \overset{\text{iid}}{\sim} \text{LogNormal}(\mu, \sigma^2 = .4) \implies Y_i \overset{\text{iid}}{\sim} \text{Normal}(\mu, \sigma^2 = .4)$$

Noninformative prior: $\pi(\mu) \propto 1$

Posterior:

$$p(\mu|\boldsymbol{y}) \propto \exp\left[\sum_{i=1}^{n} -\frac{1}{2(.4)}(y_i - \mu)^2\right]$$

Your turn

1. Write a `log_posterior` function. Notice that you can use the `dnorm` function if you log the data.

2. Run the `metropolis()` function to obtain draws from the (approximate) posterior distribution.

3. Check the trace and ACF plots to see if your chain converged and if it's working efficiently.

4. Repeat 2-3 until you're satisfied.

5. Construct and interpret a 95% credible interval for the viscosity breakdown times.