

STAT 202C - HW 3

John Rapp Farnes / 405461225

5/29/2020

1)

(a)

Let $\pi \in \text{Gamma}(4.3, 6.2)$ and proposal $g \in \text{Gamma}(4, 7)$.

For rejection sampling we want $Mg(x) \geq \pi(x) \forall x \Leftrightarrow \frac{Mg(x)}{\pi(x)} \geq 1$. We have

$$\frac{Mg(x)}{\pi(x)} = \frac{\beta_g^{\alpha_g} x^{\alpha_g-1} e^{-\beta_g x} / \Gamma(\alpha_g)}{\beta_\pi^{\alpha_\pi} x^{\alpha_\pi-1} e^{-\beta_\pi x} / \Gamma(\alpha_\pi)} = \frac{\beta_g^{\alpha_g} \Gamma(\alpha_\pi)}{\beta_\pi^{\alpha_\pi} \Gamma(\alpha_g)} x^{\alpha_g - \alpha_\pi} e^{-x(\beta_g - \beta_\pi)} \geq 1$$

TODO

```
alpha_pi <- 4.3
beta_pi <- 6.2

alpha_g <- 4
beta_g <- 7

pi_distr <- function(x) {
  dgamma(x, shape = alpha_pi, scale = beta_pi)
}

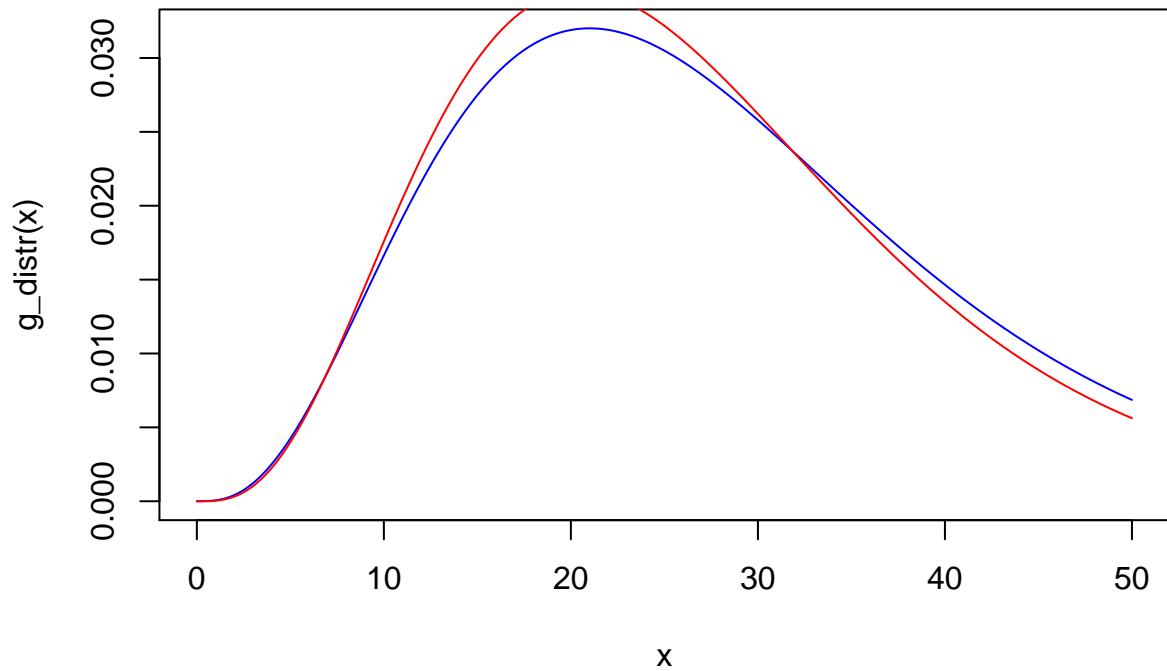
g_distr <- function(x) {
  dgamma(x, shape = alpha_g, scale = beta_g)
}

g_sample <- function(n = 1) {
  rgamma(n, shape = alpha_g, scale = beta_g)
}

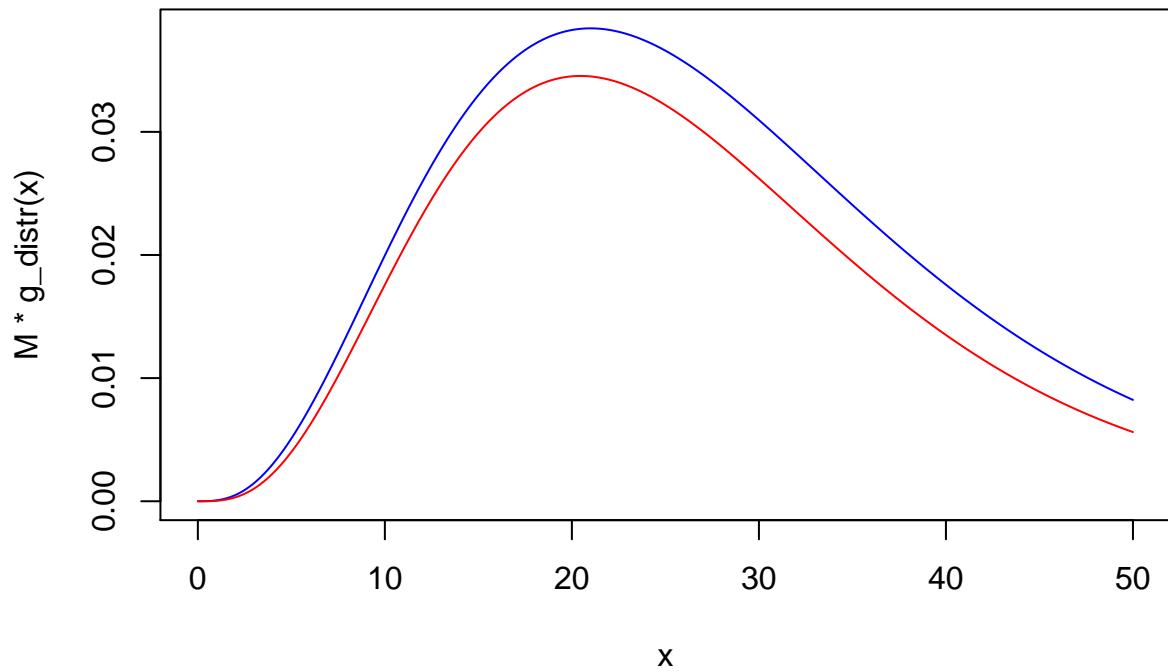
pi_div_g_distr <- function(x) {
  # TODO make more computationally efficient
  pi_distr(x) / g_distr(x)
}

x <- seq(0, 50, 0.01)

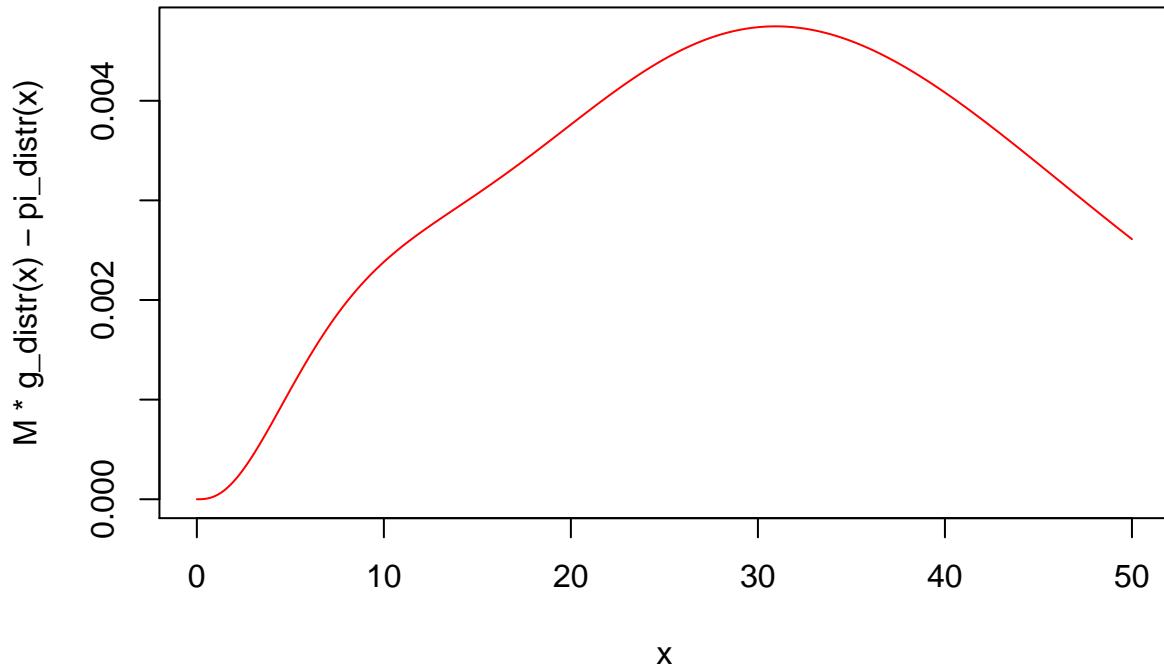
plot(x, g_distr(x), type="l", col="blue")
lines(x, pi_distr(x), col="red")
```



```
M <- 1.2  
plot(x, M*g_distr(x), type="l", col="blue")  
lines(x, pi_distr(x), col="red")
```



```
plot(x, M*g_distr(x) - pi_distr(x), type="l", col="red")
```



```

sum(M*g_distr(x) - pi_distr(x) < 0)/length(x)

## [1] 0

N <- 50000

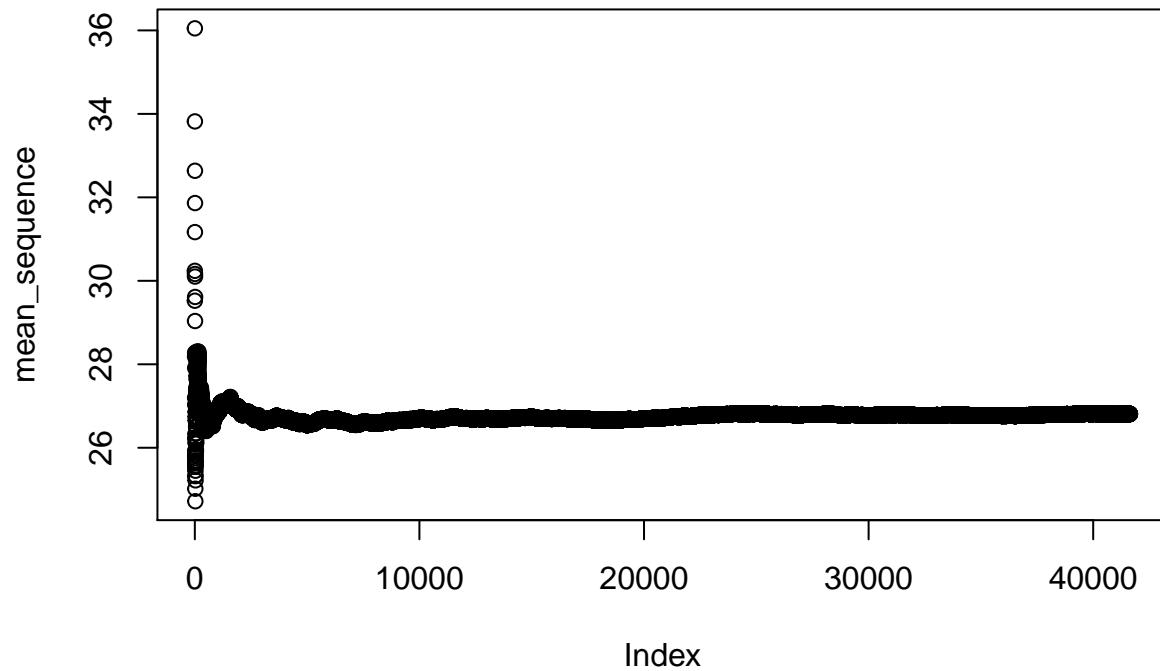
samples <- c()

# TODO optimize?
for (i in 1:N) {
  x = g_sample(1)
  r = pi_div_g_distr(x)/M

  u = runif(1)
  if (r <= u) {
    next;
  } else {
    samples <- c(samples, x)
  }
}

mean_sequence <- cumsum(samples) / seq_along(samples)
plot(mean_sequence)

```



```

mean_sequence[length(mean_sequence)]
## [1] 26.81026

mean(rgamma(N*100, shape = alpha_pi, scale = beta_pi))
## [1] 26.65786

```

(b)

```

initial_state <- 0
state <- initial_state

states <- c(state)

for (i in 1:N) {
  proposed <- g_sample()

  Q_ratio <- g_distr(proposed) / g_distr(state)
  pi_ratio <- pi_distr(proposed) / pi_distr(state)
  alpha <- Q_ratio * pi_ratio

```

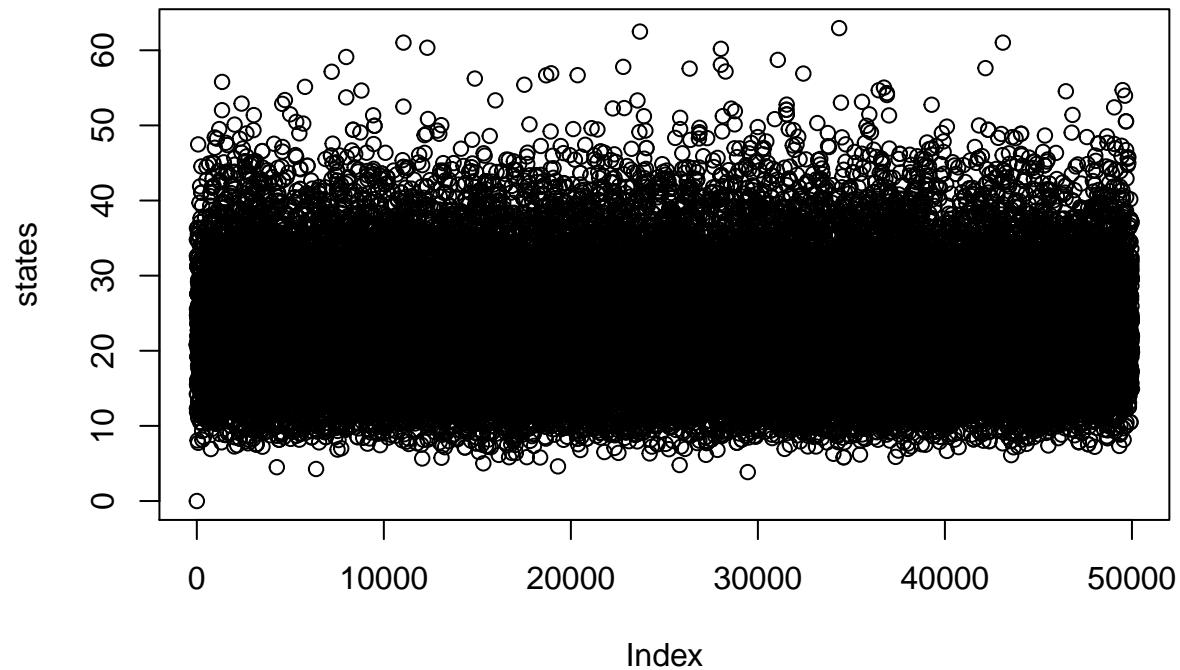
```

if (alpha >= 1 || runif(1) <= alpha) {
  state <- proposed
}

states <- c(states, state)
}

plot(states)

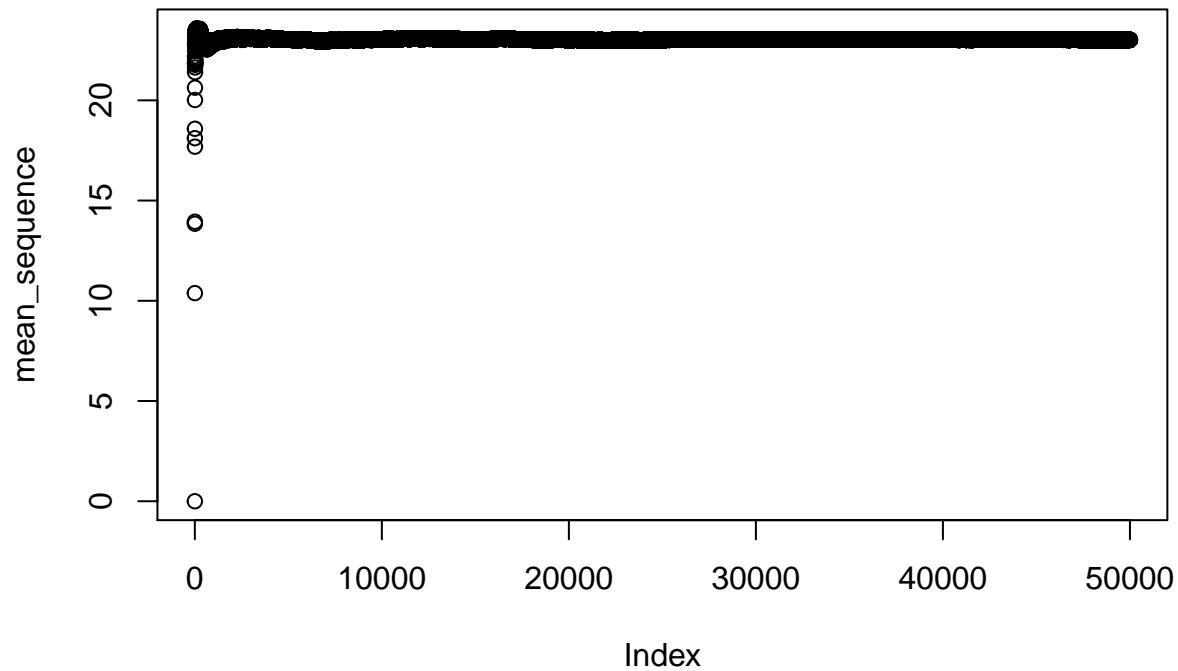
```



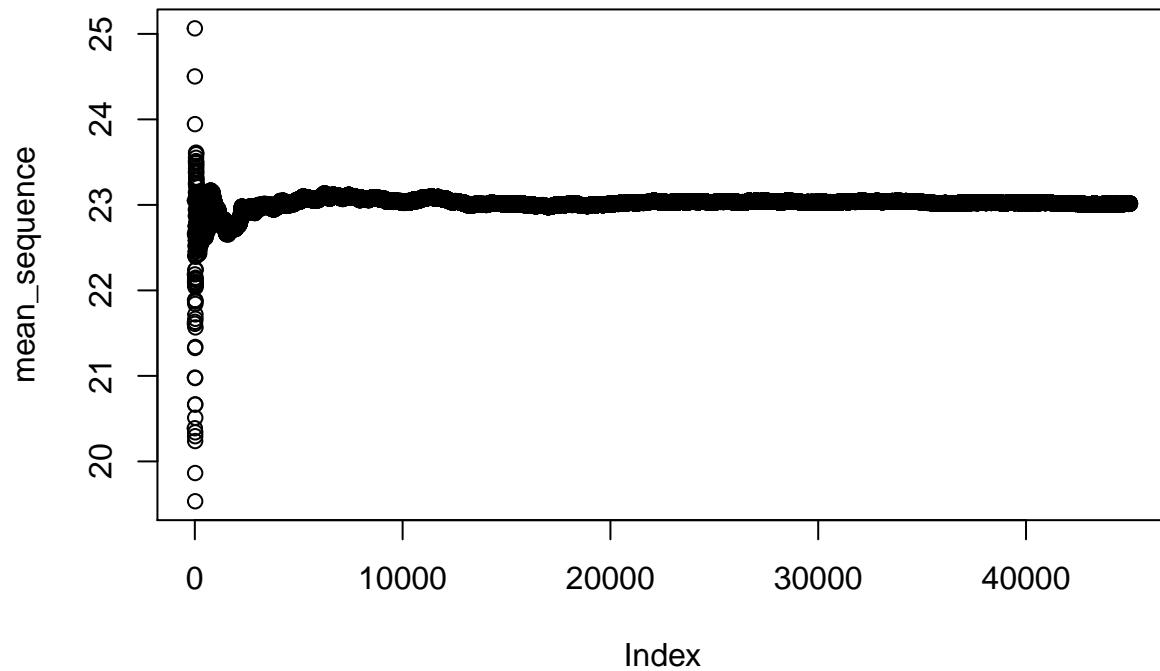
```

mean_sequence <- cumsum(states) / seq_along(states)
plot(mean_sequence)

```



```
burn_in_length <- 5000
samples <- states[burn_in_length:length(states)]
mean_sequence <- cumsum(samples) / seq_along(samples)
plot(mean_sequence)
```



```
mean_sequence[length(mean_sequence)]
```

```
## [1] 23.0146
```

(c)

```
alpha_g2 <- 5
beta_g2 <- 6

g2_distr <- function(x) {
  dgamma(x, shape = alpha_g2, scale = beta_g2)
}

g2_sample <- function(n = 1) {
  rgamma(n, shape = alpha_g2, scale = beta_g2)
}

initial_state <- 0
state <- initial_state

states <- c(state)

for (i in 1:N) {
```

```

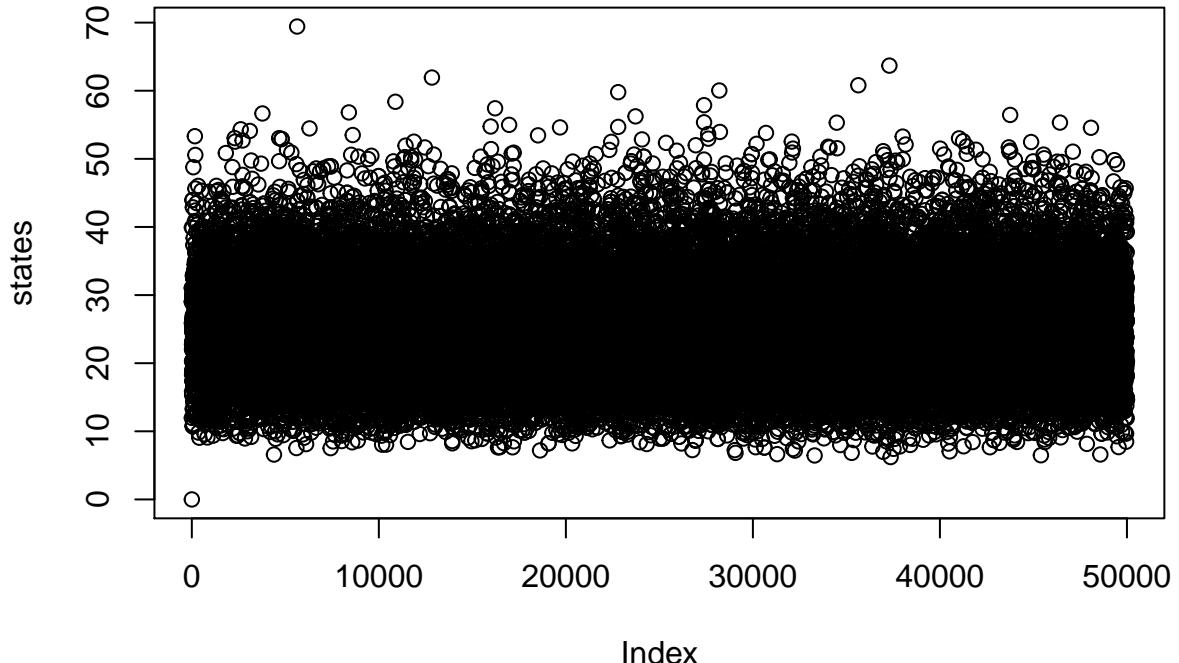
proposed <- g2_sample()
Q_ratio <- g2_distr(proposed) / g2_distr(state)
pi_ratio <- pi_distr(proposed) / pi_distr(state)
alpha <- Q_ratio * pi_ratio

if (alpha >= 1 || runif(1) <= alpha) {
  state <- proposed
}

states <- c(states, state)
}

plot(states)

```



```

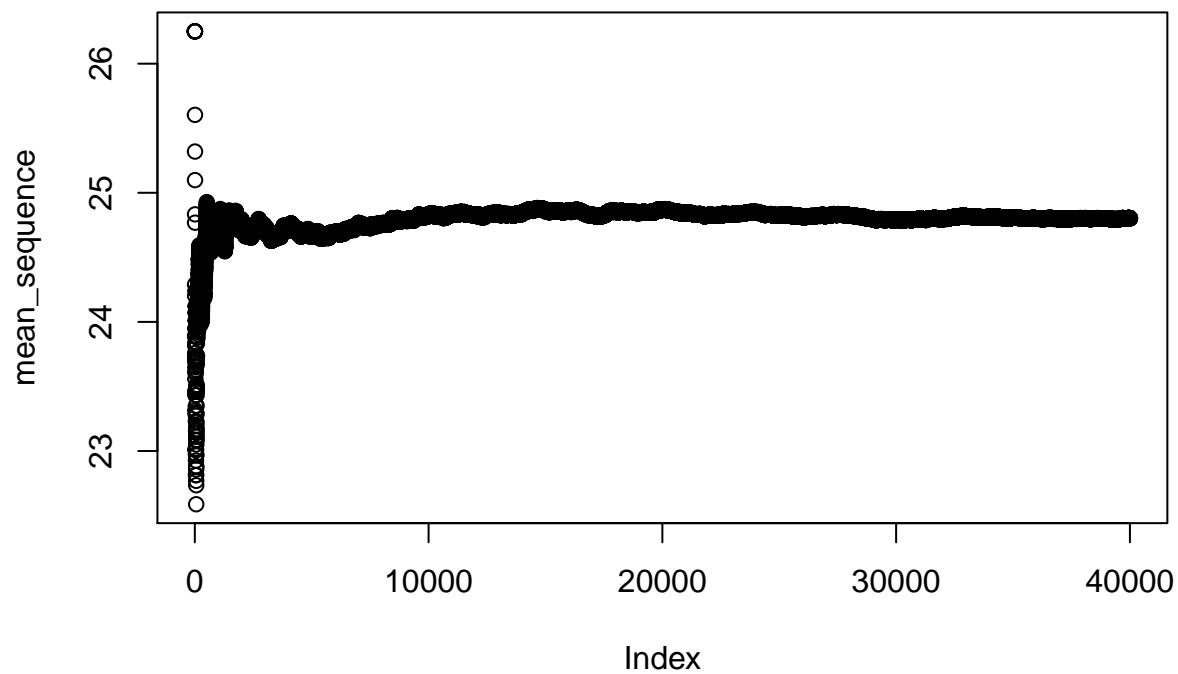
burn_in_length <- 10000

samples <- states[burn_in_length:length(states)]

mean_sequence <- cumsum(samples) / seq_along(samples)

plot(mean_sequence)

```



```
mean_sequence[length(mean_sequence)]
```

```
## [1] 24.80265
```