sta240_pset6

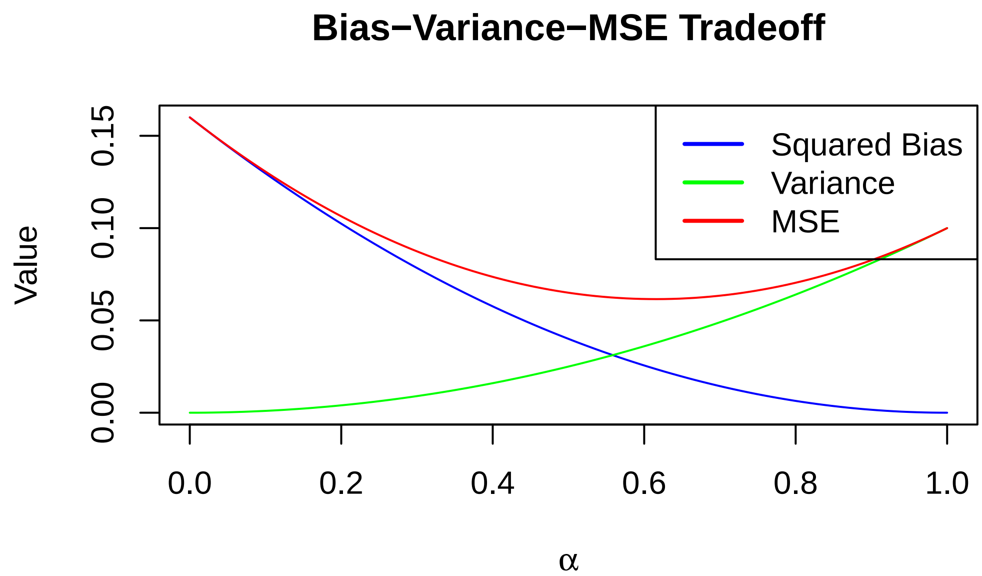
Eva Aggarwal

PSET6

Q1 c)

$$\hat{\mu}_n = \alpha \bar{X}_n + (1-\alpha)g$$

```r
mu <- 2
g <- 1.6
sigma2 <- 1
n <- 10

alpha <- seq(0, 1, length.out = 100)

bias2 <- ((alpha * mu + (1 - alpha) * g) - mu)^2
var <- (alpha^2) * (sigma2 / n)
mse <- bias2 + var

# Plot all three on the same graph
plot(alpha, bias2, type = "l", col = "blue", ylim = c(0, max(mse)),
     xlab = expression(alpha), ylab = "Value", main = "Bias-Variance-MSE Tradeoff")
lines(alpha, var, col = "green")
lines(alpha, mse, col = "red")
legend("topright", legend = c("Squared Bias", "Variance", "MSE"),
       col = c("blue", "green", "red"), lwd = 2)
```



The line graph visualizes the bias-variance trade-off:

Small alpha: estimator trusts prior too much → high bias, low variance

Large alpha: estimator trusts data → low bias, high variance

Somewhere in between is optimal: lowest MSE, best balance between bias and variance

So, the best choice of alpha— in terms of minimizing MSE — is where the red curve hits its lowest point, approx alpha = 0.6

Q2 d)

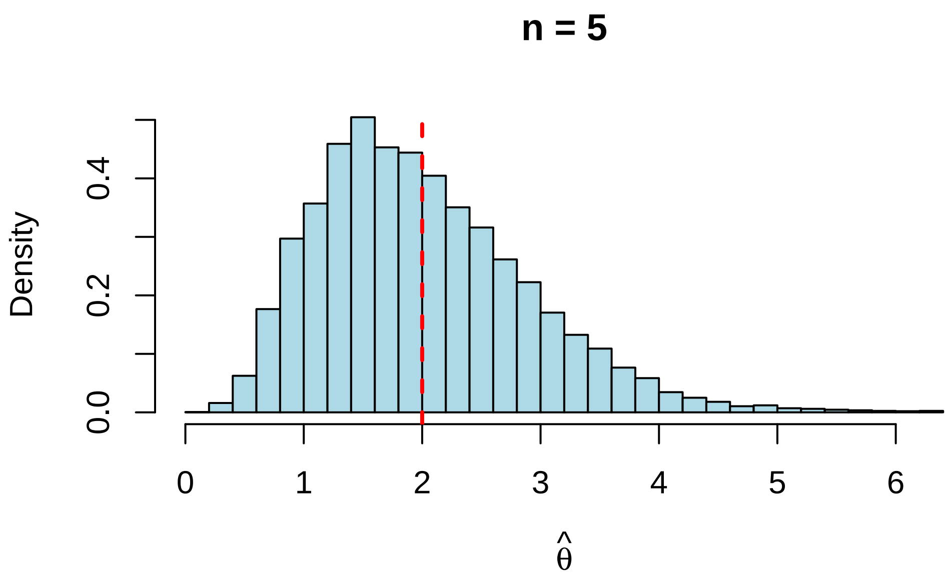
Write a function in R that takes in two arguments n and theta and returns a vector with n random numbers draw iid from the member of this family with parameter theta. In other words, fill in the blank:

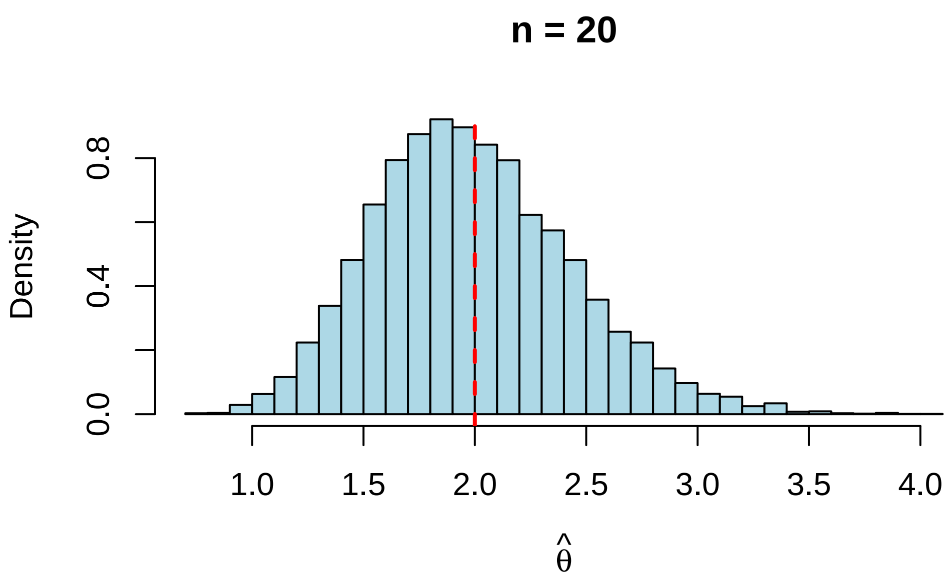
```
my_random_numbers <- function(n, theta){
  y <- rexp(n, 1/(2*theta))
  x <- sqrt(y)
  return(x)
}
```

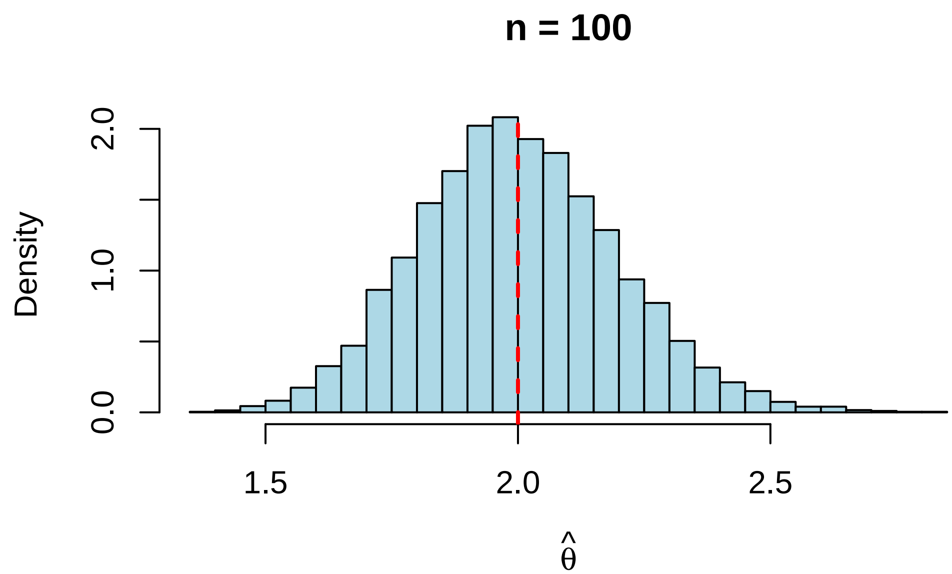
Q2 e)

The histograms demonstrate that $\hat{\theta}_n$ is an unbiased (centers around true value in diff sample sizes), consistent (converges to true value as n increases) estimator with decreasing variance as sample size increases. Asymptotic normality can be seen by the increasingly bell-shaped distribution for large n.

```r
simulate_mle <- function(n, theta, reps = 10000) {
  replicate(reps, {
    x <- my_random_numbers(n, theta)
    mean(x^2) / 2
  })
}

theta_true <- 2
sample_sizes <- c(5, 20, 100)
set.seed(123)

for (n in sample_sizes) {
  samples <- simulate_mle(n, theta_true)

  hist(samples,breaks = "Scott",
       col = "lightblue",
       main = paste("n =", n),
       xlab = expression(hat(theta)),
       freq = FALSE)
  abline(v = theta_true, col = "red", lwd = 2, lty = 2)
}
```


n = 5
Density
0.0
0.2
0.4
0
1
2
3
4
5
6
$\hat{\theta}$


n = 20
Density
0.0
0.4
0.8
1.0
1.5
2.0
2.5
3.0
3.5
4.0
$\hat{\theta}$



Q3 d)

She's 98.3% confident that $\lambda \leq 3$ and that $\frac{\alpha_0}{\beta_0} = E[X] \approx 1$

```
prior <- function(mean_guess, upper_bound, upper_prob = 0.983) {
  f <- function(alpha) pgamma(upper_bound, shape = alpha, rate = alpha / mean_guess) - upper_
  alpha0 <- uniroot(f, lower = 0.1, upper = 100)$root
  beta0 <- alpha0 / mean_guess
  return(list(alpha0 = alpha0, beta0 = beta0))
}
prior(mean_guess = 1, upper_bound = 3)
```

```
$alpha0
[1] 2.019699

$beta0
[1] 2.019699
```

Hyperparameters should be set such that it is centered around 1, but still leaves room (although slight) to be 1 or 2.