a.

```
moments_method <- function(x) {
  m <- mean(x)
  if (m == 1.6) {
    return(NA)
  }
  return(0.6/(m - 1.6))
}
```

b.

```
gen_sample <- function(theta) {
  n <- 1000000
  p1 <- 0.6
  lam <- 0.25

  temp <- rbinom(n, size=1, prob=p1)
  sample <- ifelse(
    temp,
    rexp(n, rate=theta),
    rexp(n, rate=lam)
  )
  return(sample)
}

pars <- c(0.5, 1, 1.5, 2, 2.5, 3, 5, 10, 20)
res <- numeric(length(pars))
for (i in 1:length(pars)) {
  res[i] <- moments_method(gen_sample(pars[i]))
}
print(pars)
```

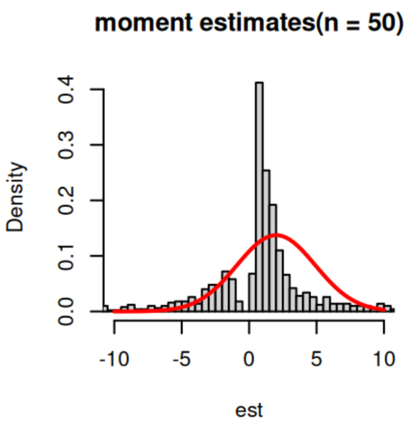
```
## [1]  0.5  1.0  1.5  2.0  2.5  3.0  5.0 10.0 20.0
```

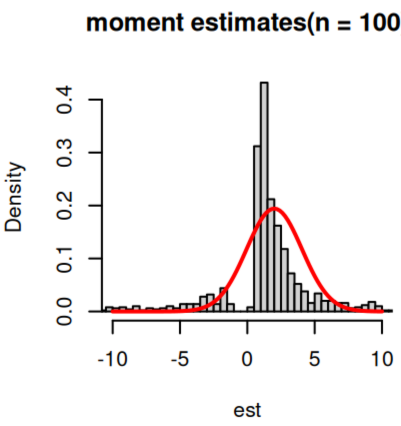
```
print(res)
```

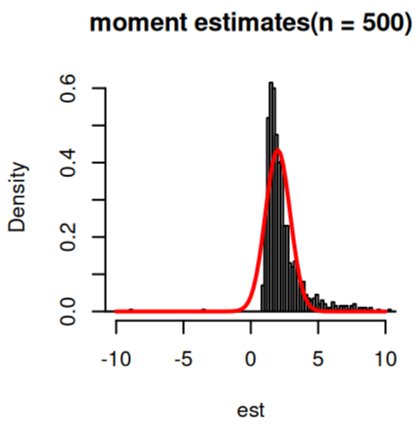
```
## [1]  0.4981117  1.0019406  1.5322512  1.9937246  2.5250714  3.0790361  4.8382045
## [8] 10.2651251 20.1923284
```

c.

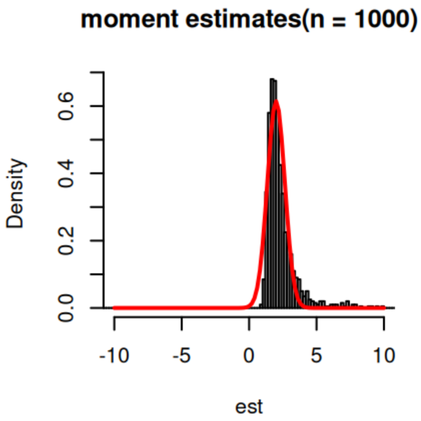
```
gen_sample_n <- function(n, theta) {
  p1 <- 0.6
  lam <- 0.25

  temp <- rbinom(n, size=1, prob=p1)
  sample <- ifelse(
    temp,
    rexp(n, rate=theta),
    rexp(n, rate=lam)
  )
  return(sample)
}

asymp_compare <- function(n, theta) {
  k <- 1000
  est <- numeric(k)
  for (i in 1:k) {
    est[i] <- moments_method(gen_sample_n(n, theta))
  }
  est <- na.omit(est)
  if (n > 1000) {
    hist(est,
         main=sprintf("moment estimates(n = %d)", n),
         xlim=c(1,3),
         breaks="FD",
         freq=FALSE)
  }
  else {
    hist(est,
         main=sprintf("moment estimates(n = %d)", n),
         xlim=c(-10,10),
         breaks="FD",
         freq=FALSE)
  }
  curve(dnorm(x, mean = theta, sd = sqrt(3796 / (9 * n))),
        add = TRUE,
        col = "red",
        lwd = 2)
}

n_values <- c(50, 100, 500, 1000, 10000)
theta <- 2

par(mfrow=c(2, 3))
for (n in n_values) {
  asymp_compare(n, theta)
}
```

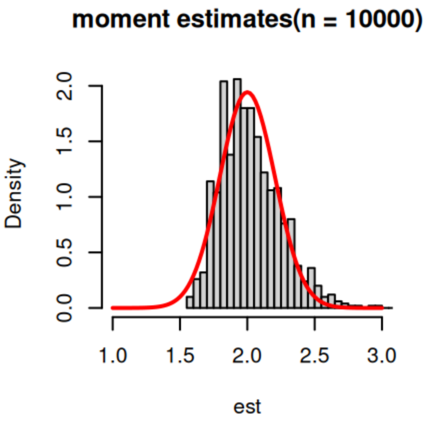






d)





```
theor_var <- function(theta) {
  return((256*theta^4-48*theta^3+21*theta^2)/9)
}

conf_int <- function(sample, n) {
  est <- moments_method(sample)
  left <- est - 1.96*sqrt(theor_var(est)/n)
  right <- est + 1.96*sqrt(theor_var(est)/n)
  return(c(left, right))
}

n_values <- c(50, 100, 500, 1000, 10000)
theta <- 2

for (n in n_values) {
  sample <- gen_sample_n(n, theta)
  print(conf_int(sample, n))
}
```

```
## [1] -101.22485   85.43185
## [1] -438.7649  480.8020
## [1] 0.3785379 3.1164165
## [1] 0.7909509 2.2846691
## [1] 1.700141 2.658215
```