Monte Carlo Simulation

Devere Anthony Weaver

2022-07-02

Contents

Monte Carlo Simulation

When we generate a sample using a compute program, we already know the population parameters. We can apply the same estimators to this artificial sample to estimate the population parameters.

Now, this gets a little weird, but just stick with it. The process is: 1. Select a population distribution and its parameters 2. Generate a sample from this distribution 3. Use this sample to estimate the population parameters

Again, this is strange, but the goal here is to obtain a noisy estimate of something we know exactly so that we are able to study the behavior of our estimator very well.

This method is often used in research since it provides the only way to learn about important features of the estimators and statistical tests.

Finite Sample Properties of Estimators

This section is a look at an example where we want to simulate a situation in which we want to estimate the mean of a normally distributed random variable, $Y \sim N(\mu, \sigma^2)$, using a sample size n.

The obvious estimator for the population mean is our sample mean, but the question becomes "what properties does this estimator have?". We know from intro stats classes that the sample mean is normally distributed. We can use simulation to verify.

In the simulation, we take a sample size of $n = 100$ from a population distribution with $\mu = 10$ and $\sigma = 2$. The we calculate the sample mean as an estimate of μ and see the result for three different samples. In other words, we are just simply calculating the sample means of three different samples from the sample population in order to verify that the sample mean is a good estimator for the population mean when the population distribution has these given parameters.

```
# set random seed for reproducible results
set.seed(123456)
```

```r
# draw first random sample of 100 from population with given parameters above
sample1 <- rnorm(100, 10, 2)

# estimate the population mean with the sample mean
mean(sample1)
```

```
## [1] 10.03364
```

```r
# draw second sample from same population
sample2 <- rnorm(100, 10, 2)
mean(sample2)
```

```
## [1] 9.913197
```

```r
# draw third sample
sample3 <- rnorm(100, 10, 2)
mean(sample3)
```

```
## [1] 10.21746
```

As you can see, all the sample means are very close to the population mean, which we know. The small deviations from the population mean is simply the nature of sampling noise.

So, while we only used three samples here, in reality a *good* Monte Carlo simulation should use as many samples as possible. To address this, we can repeat the above simulation, but instead of three samples, we'll use 10,000. We'll then have a vector containing the sample means for each of the 10,000 samples.

```r
set.seed(123456)

r <- 10000
ybar <- numeric(r)

# take 10,000 samples and compute their means
for(i in 1:r){
  sample <- rnorm(100, 10, 2)
  ybar[i] <- mean(sample)
}
```

Now that we've gotten our vector with the 10,000 sample means, we can use this to confirm that the average of these repeated sample means is indeed close to the mean of the population they were drawn from. We can also use a kernel density estimator to compare the sample means and sample variances to the theoretical normal distribution.
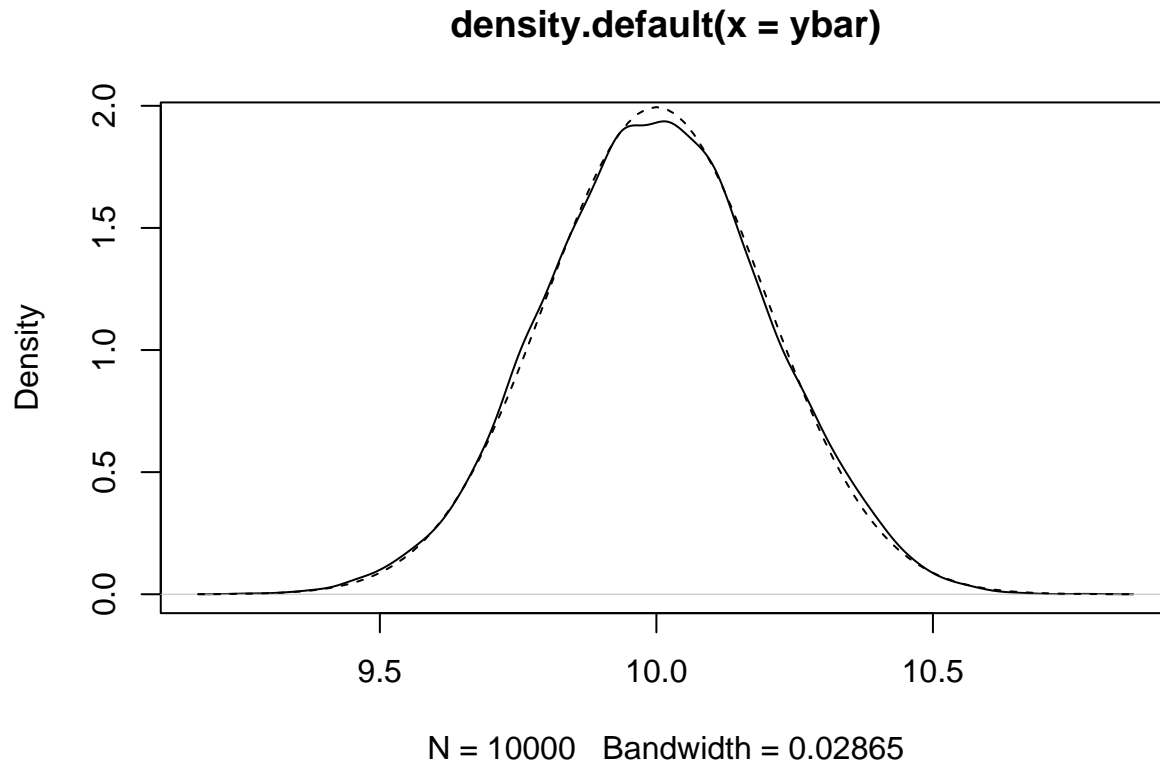
```r
# compute mean of the 10,000 sample means
mean(ybar)
```

```
## [1] 9.998861
```

```r
# compute the variance of the 10,000 sample means
var(ybar)
```

```
## [1] 0.04034146
```

```r
# simulate density
plot(density(ybar))
curve(dnorm(x, 10, sqrt(.04)), add=T, lty=2)
```

density.default(x = ybar)

N = 10000   Bandwidth = 0.02865

These results show that the sample mean is indeed a good estimator for a population mean. The mean of the sample means ends up being *very* close to the population mean of 10, with the expected variance. This is a fairly simple example; however, for more advanced estimators, a simulation such as this is the only real way to study their properties since it's impossible to derive theoretical results of interest.

Of course, this was simply a brief introduction to Monte Carlo simulation. As such, be sure to do a deeper dive into this kind of simulation.

Asymptotic Properties of Estimators

Using Monte Carlo simulation as demonstrated above, we can see that for our estimator, sample mean, as the sample size tends toward infinity, the expected value of the sample mean will converge to the population mean (law of large numbers).

Using the same code from above, increasing the sample size and then graphically examining it will show that as the sample size increases, the variance decreases. This means that with a large and larger sample size, the sample mean will get closer and closer to the actual mean. Visually, this results in a density plot that becomes taller and thinner since the variance decreases.

Even though it is theoretically possible, it is physically impossible for a computer to go to infinity. However, increasing the sample size will show that eventually the density plot will converge to a vertical line at which is exaclty the same value as the population mean.