Lab 1: Monte Carlo Simulation

2025-01-10

We use simulation to understand properties of random variables

A very common computational tool used to obtain numerical results instead of analytical solutions is Monte Carlo simulation. The key idea is that we can approximate the properties of random variable X (e.g., mean, variance) by taking a repeated number of i.i.d. draws from that random variable and computing them from the empirical distribution.

- Data-generating process
- Iterations
- Random seed

Illustration: Bias of the uncorrected variance estimator

Let's show the analytical solutions we got from mathematical proofs via simulation! Start by defining the data-generating process for a fixed value of $N = 30$, $\mu = 0$, $\sigma^2 = 4$, We'll assume X is normal, though this isn't strictly necessary.

```r
# Set random seed
set.seed(60637)

# Define the parameters of the DGP
N = 30
mu = 0
sigma = sqrt(4)

# Define a function to compute the unadjusted variance
var_unadj = function(x){
  return((1/length(x))*sum((x - mean(x))^2))
}

# Number of iterations
niter = 10000

# Run the simulation and get evaluations of the unadjusted variance
sim_results = 1:niter %>% map_dbl(function(x) var_unadj(rnorm(n=N, mean=mu, sd=sigma)))

# Calculate bias
mean(sim_results) - sigma^2
```

```
## [1] -0.1172675
```

Now, we can show that this bias goes away when we use the $n-1$ correction (also called "Bessel's Correction").

```r
# Set random seed
set.seed(60637)

# Define the parameters of the DGP
N = 30
```

```r
mu = 0
sigma = sqrt(4)

# Define a function to compute the unadjusted variance
# This function is equivalent to var() in R
var_adj = function(x){
  return((1/(length(x)-1))*sum((x - mean(x))^2))
}

# Number of iterations
niter = 10000

# Run the simulation and get evaluations of the unadjusted variance
sim_results_adj = 1:niter %>% map_dbl(function(x) var_adj(rnorm(n=N, mean=mu, sd=sigma)))

# Calculate bias
mean(sim_results_adj) - sigma^2
```

```
## [1] 0.01661985
```

In-Class Exercise: Sampling variance of the difference-in-means estimator for the ATE

Data-generating process:

- The outcome is generated by $Y_i = \tau D_i + \epsilon_i$ where $\epsilon_i \sim Normal(0, 4)$
- Treatment assignment mechanism: half receive treatment randomly
- Numbers of random draws in each iteration: 30

Iterations: 10000

Random seed: 60637