Simulation Studies

Last modified: 2026-11-13 00:22:44 (PST)

Theory often cannot give the bias, variance or coverage of an estimator when the estimator or the data distribution is complicated. A simulation study measures the same properties for any estimator a computer can run. It generates many data sets from a distribution whose parameters are known, applies the estimator to each, and summarizes the results.

This page defines the parts of a simulation study, shows how to measure the study's own accuracy, and works one example in R and Python whose answer is known.

The data-generating process

Definition 0.1 (Data-generating process)**.** The **data-generating process (DGP)** of a simulation is the joint probability distribution F_{θ_0} from which it draws its data $X_1, \dots, X_n$, with the number of observations n and the parameter value θ_0 fixed and known:

$$X_1, \dots, X_n \sim F_{\theta_0}.$$

Example 0.1 (A normal data-generating process)**.**
Let $X_1, \dots, X_n$ be IID normal random variables with expectation $\mu = 0$ and variance $\sigma^2 = 4$, with $n = 10$. The distribution of these variables is a data-generating process, and the estimand $\sigma^2 = 4$ is known exactly, so the error of any estimator of σ^2 can be computed.

Replications and simulation studies

Definition 0.1 (Replication)**.** A **replication** of a simulation is one run of its steps: draw a data set $X_1, \dots, X_n^{(m)}$ from the data-generating process, then compute the estimate $T_m \stackrel{\text{def}}{=} \hat{\theta}\left(X_1, \dots, X_n^{(m)}\right)$, where $\hat{\theta}$ is an estimator, for $m = 1, \dots, M$. Different replications draw their data sets independently.

Example 0.1 (A replication of the normal data-generating process)**.**
For the normal data-generating process, one replication draws $n = 10$ values from the normal distribution with expectation 0 and variance 4. It then computes the estimate T_m of the variance σ^2 from those 10 values. A second replication draws 10 new values, independently of the first, and computes T_{m+1}.

Definition 0.2 (Simulation study)**.** A **simulation study** runs M replications of one data-generating process and one estimator, then summarizes the estimates $T_1, \dots, T_M$ to learn the estimator's sampling distribution.

Example 0.2 (A simulation study of a variance estimator)**.**
For the normal data-generating process, a simulation study of one estimator of the variance σ^2 runs M replications of that process and that estimator. The M estimates $T_1, \dots, T_M$ then show how the estimator behaves across repeated data sets.

Simulation estimates

Definition 0.1 (Simulation estimate)**.** A **simulation estimate** of the expectation $\mathbb{E}[T]$ of an estimate T is the average of the replications' estimates $T_1, \dots, T_M$:

$$\bar{T}_M \stackrel{\text{def}}{=} \frac{1}{M} \sum_{m=1}^{M} T_m.$$

Example 0.1 (A simulation estimate from three replications)**.**
Suppose $M = 3$ replications give the estimates $T_1 = 3$, $T_2 = 5$ and $T_3 = 4$. Then the simulation estimate of $\mathbb{E}[T]$ is

$$\bar{T}_3 = \frac{3 + 5 + 4}{3} = 4.$$

Definition 0.2 (Simulation estimate of the bias)**.** Let θ be an estimand, let $\hat{\theta}$ be an estimator, and let $T_1, \ldots, T_M$ be its estimates from M replications. The **simulation estimate of the bias** of $\hat{\theta}$ is the simulation estimate $\bar{T}_M$ of $\mathbb{E}\left[\hat{\theta}\right]$, minus θ:

$$\bar{T}_M - \theta.$$

Example 0.2 (A simulation estimate of the bias from three replications)**.**
Continue the three replications, and let the estimand be $\theta = 5$. Then the simulation estimate of the bias is

$$\bar{T}_3 - \theta = 4 - 5 = -1.$$

Definition 0.3 (Simulation estimate of the mean squared error)**.** Let θ be an estimand, let $\hat{\theta}$ be an estimator, and let $T_1, \ldots, T_M$ be its estimates from M replications. The **simulation estimate of the mean squared error** of $\hat{\theta}$ is the average of the squared differences $(T_m - \theta)^2$:

$$\frac{1}{M} \sum_{m=1}^{M} (T_m - \theta)^2.$$

Example 0.3 (A simulation estimate of the mean squared error from three replications)**.**

Continue the three replications with $\theta = 5$. The squared differences are $(3-5)^2 = 4$, $(5-5)^2 = 0$ and $(4-5)^2 = 1$, so the simulation estimate of the mean squared error is

$$\frac{4+0+1}{3} = \frac{5}{3}.$$

Remark (Simulation estimates in Bayesian inference)

Remark 0.1 (Simulation estimates in Bayesian inference). The Monte Carlo estimate of a posterior expectation is the same average, taken over draws from a posterior distribution instead of over replications of a data-generating process.

Monte Carlo error

Definition 0.1 (Monte Carlo standard error)**.** The **Monte Carlo standard error (MCSE)** of a simulation estimate $\bar{T}_M$ is its standard error over repeated simulation studies:

$$\mathrm{SE}(\bar{T}_M) = \mathrm{SD}(\bar{T}_M).$$

Theorem 0.1 (Monte Carlo standard error of a simulation estimate)**.** *Let $T_1, \ldots, T_M$ be the estimates of M replications of a simulation study, which are IID with expectation $\mathbb{E}[T]$ and finite variance $\mathrm{Var}(T)$. Then the simulation estimate $\bar{T}_M$ is unbiased for $\mathbb{E}[T]$, and its Monte Carlo standard error is*

$$\mathrm{SE}(\bar{T}_M) = \frac{\mathrm{SD}(T)}{\sqrt{M}}.$$

Proof

Proof. By the mean and variance of a sum of IID random variables, the sum $S_M \overset{\text{def}}{=} \sum_{m=1}^{M} T_m$ has $\mathbb{E}[S_M] = M\,\mathbb{E}[T]$ and $\mathrm{Var}(S_M) = M\,\mathrm{Var}(T)$. Since $\bar{T}_M = S_M/M$:

$$\begin{aligned}
\mathbb{E}\big[\bar{T}_M\big] &= \frac{\mathbb{E}[S_M]}{M} && \text{(constants factor out of expectations)} \\
&= \frac{M\,\mathbb{E}[T]}{M} && \text{(substitute } \mathbb{E}[S_M]\text{)} \\
&= \frac{M}{M}\,\mathbb{E}[T] && \text{(regroup the factors)} \\
&= 1 \cdot \mathbb{E}[T] && (M/M = 1) \\
&= \mathbb{E}[T] && \text{(multiplying by 1 changes nothing)}
\end{aligned}$$

For the variance:

$$
\begin{aligned}
\operatorname{Var}(\bar{T}_M) &= \frac{\operatorname{Var}(S_M)}{M^2} && \text{(a constant factor } c \text{ scales variance by } c^2\text{)} \\
&= \frac{M \operatorname{Var}(T)}{M^2} && \text{(substitute } \operatorname{Var}(S_M)\text{)} \\
&= \frac{M \operatorname{Var}(T)}{M \cdot M} && \text{(write } M^2 \text{ as } M \cdot M\text{)} \\
&= \frac{M}{M} \cdot \frac{\operatorname{Var}(T)}{M} && \text{(split the fraction into two factors)} \\
&= 1 \cdot \frac{\operatorname{Var}(T)}{M} && (M/M = 1) \\
&= \frac{\operatorname{Var}(T)}{M} && \text{(multiplying by 1 changes nothing)} \\
\operatorname{SE}(\bar{T}_M) &= \sqrt{\operatorname{Var}(\bar{T}_M)} && \text{(standard error is the standard deviation)} \\
&= \sqrt{\frac{\operatorname{Var}(T)}{M}} && \text{(substitute } \operatorname{Var}(\bar{T}_M)\text{)} \\
&= \frac{\sqrt{\operatorname{Var}(T)}}{\sqrt{M}} && \text{(the square root of a quotient is the quotient of square roots)} \\
&= \frac{\operatorname{SD}(T)}{\sqrt{M}} && \text{(the standard deviation is the square root of the variance)}
\end{aligned}
$$

□

Remark (Reporting a Monte Carlo standard error)

Remark 0.1 (Reporting a Monte Carlo standard error). The standard deviation $\operatorname{SD}(T)$ in Theorem 0.1 is unknown, so a study reports the sample standard deviation s_T of the estimates $T_1, \ldots, T_M$ divided by $\sqrt{M}$ in its place. That Monte Carlo standard error shrinks like $1/\sqrt{M}$, so quadrupling the number of replications halves it. A simulation estimate should be reported with its Monte Carlo standard error, so a reader can tell a real difference from random variation in the simulation.

Seeds

Definition 0.1 (Seed). A **seed** is the integer that sets the starting state of a computer's pseudorandom number generator, so that rerunning a simulation with the same seed reproduces the same random draws:

$$\text{seed} \mapsto U_1, U_2, \ldots$$

Remark (Setting seeds)

Remark 0.1 (Setting seeds). Set the seed once, at the top of the simulation, and record its value with the results. Different software uses different generators, so the same seed gives different draws in R and in Python. Both give correct simulation estimates, which agree up to their Monte Carlo standard errors.

A worked simulation study

Example 0.1 (Simulating the bias of the divide-by-n variance estimator)**.** For the data-generating process with $\mu = 0$, $\sigma^2 = 4$ and $n = 10$, we simulate the bias of the divide-by-n estimator $\hat{\sigma}^2$ of σ^2 from the example of a biased variance estimator. That example shows that the bias is exactly $-\sigma^2/n = -0.4$, so the simulation can be checked against the known answer.

Each replication draws $n = 10$ values and computes one estimate of σ^2. We run $M = 10000$ replications, set the seed once at the start, and report the simulation estimate of the bias with its Monte Carlo standard error. The choice $M = 10000$ gives a Monte Carlo standard error of about 0.02, small next to the bias.

R

```r
n <- 10
sigma_sq <- 4
n_reps <- 10000
n_draws <- n_reps * n
sigma <- sqrt(sigma_sq)

set.seed(571)
draws <- rnorm(n_draws, mean = 0, sd = sigma)
# Each row is one replication's data set.
x <- matrix(draws, nrow = n_reps)

x_bar <- rowMeans(x)
deviations <- x - x_bar
squared_deviations <- deviations^2
var_hat <- rowMeans(squared_deviations)

bias_hat <- mean(var_hat) - sigma_sq
mc_se <- sd(var_hat) / sqrt(n_reps)

c(
  simulation_estimate = bias_hat,
  mc_standard_error = mc_se,
  known_bias = -sigma_sq / n
)
#> simulation_estimate   mc_standard_error          known_bias
#>          -0.3915413           0.0169017          -0.4000000
```

Python

```python
import numpy as np

n = 10
sigma_sq = 4
n_reps = 10000

rng = np.random.default_rng(571)
# Each row is one replication's data set.
x = rng.normal(loc=0, scale=np.sqrt(sigma_sq), size=(n_reps, n))

x_bar = x.mean(axis=1, keepdims=True)
var_hat = ((x - x_bar) ** 2).mean(axis=1)

bias_hat = var_hat.mean() - sigma_sq
mc_se = var_hat.std(ddof=1) / np.sqrt(n_reps)

print(f"simulation estimate: {bias_hat:.3f}")
#> simulation estimate: -0.390
print(f"mc standard error:   {mc_se:.3f}")
#> mc standard error:   0.017
print(f"known bias:          {-sigma_sq / n:.3f}")
#> known bias:          -0.400
```

The R simulation estimate of the bias is -0.392, with a Monte Carlo standard error of 0.017. It is within 0.5 Monte Carlo standard errors of the known bias -0.4. The Python run uses a different random number generator, so its estimate differs slightly, but it too agrees with -0.4 up to its Monte Carlo standard error.

Figure 1 shows the sampling distribution of $\hat{\sigma}^2$ that the replications trace out. Its histogram uses 40 equal-width bins, which put about 250 of the 10^4 estimates in the average bin. Fewer bins, such as 10, would hide the right skew of the distribution in a few wide bars. More bins, such as 200, would leave about 50 estimates per bin on average, and random bar-to-bar variation would hide the shape. The bin count changes only the picture, not any number reported above.

```r
estimates <- tibble::tibble(var_hat = var_hat)
mean_var_hat <- mean(var_hat)
line_values <- c(sigma_sq, mean_var_hat)
line_labels <- c("True variance", "Average estimate")
means <- tibble::tibble(value = line_values, line = line_labels)

estimates |>
  ggplot() +
  aes(x = var_hat) +
  # check-unexplained-choices: allow
  geom_histogram(bins = 40, fill = "grey80", color = "grey40") +
  geom_vline(
    data = means,
    mapping = aes(xintercept = value, linetype = line)
  ) +
  labs(x = "Estimate of the variance", y = "Replications", linetype = NULL)
```

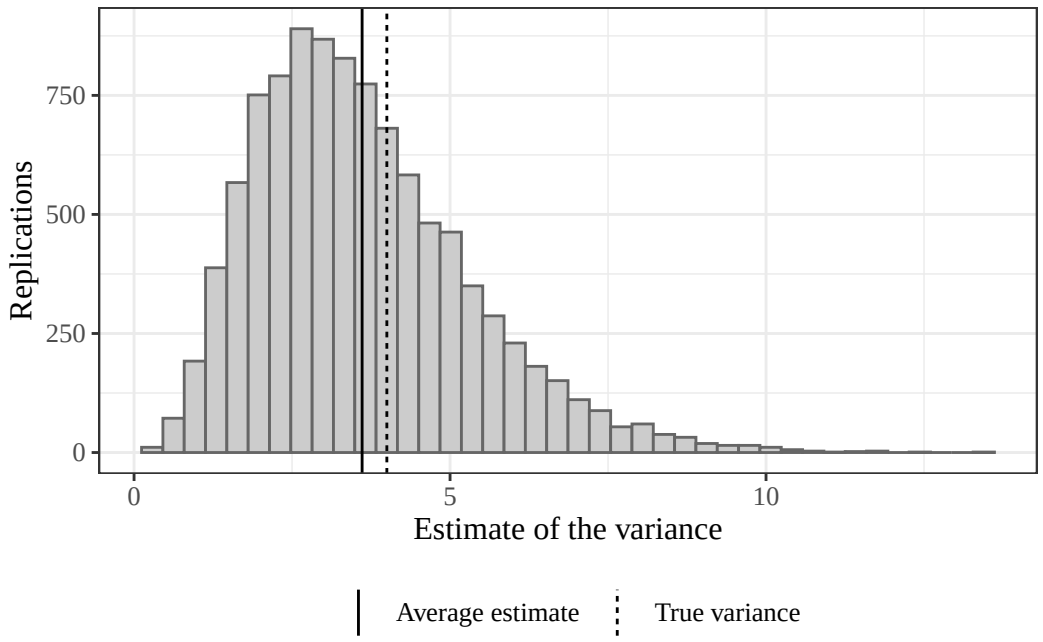


Figure 1: Estimates of the variance from 10,000 replications. The vertical lines mark the true variance 4 and the average estimate, which falls below it by the bias.

The distribution is right-skewed, and its average sits below the true variance, which is the bias.

References