

Variance Correction Factors: Analysis, Derivation, and Applications

Introduction

In statistical inference, estimating the dispersion of a population from a finite sample is foundational. The choice of the divisor directly alters the properties of the variance estimator, exposing a fundamental trade-off between unbiasedness and minimum variance. This report comprehensively evaluates multiple distinct divisors: $n - 2$, $n - 1.5$, $n - 1$ (Bessel's correction), $n - 0.66$, $n - 0.5$, and $n + 1$.

Because the sample data is naturally closer to its own mean than to the population mean, a naive calculation of variance (dividing by the total number of observations, n) systematically underestimates the true population variance. This underestimation is a mathematical bias. To correct this bias, or to optimize the estimate for other properties like Mean Squared Error (MSE), statisticians apply various correction factors, or divisors, to the variance formula. The choice of divisor fundamentally alters the characteristics of the estimator, determining whether it is unbiased, minimum-MSE, or optimized for standard deviation.

History and Context of Bessel's Correction

The standard practice of dividing by $n - 1$ instead of n is called Bessel's correction, named after Friedrich Wilhelm Bessel, a German mathematician and astronomer. In 1815, while analyzing observational errors in astronomical data, Bessel noted that using the sample mean rather than the true population mean caused a systematic underestimation of the error spread.

Carl Friedrich Gauss also contributed to this concept in 1821, showing that when data points are forced to benchmark against a self-derived mean, the deviations automatically squeeze closer together. The mathematical necessity arises because a sample mean always shifts to the exact center of its unique cluster. Consequently, the squared distances to the sample mean are smaller than the squared distances to the true, hidden population mean. Dividing by $n - 1$ perfectly offsets this variance

compression, shifting the mean of the distribution rightward to align with the true population parameters.

You can view the full mathematical [derivation](#) in the appendix.

Experiment Setup

To benchmark the performance of different correction factors, I ran a Monte Carlo Simulation in Python. I set up a synthetic population with a given mean and standard deviation and use Numpy to select from both normal and lognormal distributions with those parameters. I selected samples with $n=10$ so that the performance gap between the estimators would be very apparent, since it converges as n approaches infinity. I took 100,000 samples from both the normal and lognormal distributions and calculated the sample variances for every sample using each different correction factor then graphed them. Finally, I compared the average sample variance using each correction factor to the true population variance as well as calculating the MSE for each correction factor. The resulting [graphs](#), [tabular data](#), and [code](#) can be found in the appendix.

Granular Application of Estimators

Divisor $n - 2$ (Over-correction): This divisor expands the output estimate heavily. It is used when multiple structural parameters have been exhausted beforehand (such as adjusting degrees of freedom in simple linear regressions where both intercept and slope are pre-calculated). In standalone variance tracking, it is too unstable, leading to a massive MSE explosion.

Divisor $n - 1.5$ (Unbiased σ): Due to the Jensen's Inequality, for any nonlinear concave function (such as the square root) we have $E[S] \neq \sqrt{E[S^2]}$. This means that if we want to unbiased the standard deviation instead of the variance we should use a correction factor of $n-1.5$

Divisor $n - 1$ (Unbiased σ^2): Used when your priority is avoiding long-term systematic bias. It forms the standard baseline for hypothesis testing, confidence interval modeling, and regulatory safety benchmarks because it guarantees long-run convergence across separate independent labs.

Divisors $n - 0.66$ and $n - 0.5$ (Hybrid Structures): These fractional values bridge the gap between unbiasedness and error minimization. Specifically, $n - 0.5$ is deployed for median-unbiased estimation or within specialized non-parametric frameworks.

Divisor $n + 1$ (Minimum MSE Optimization): Used when minimizing individual point-prediction error is paramount. It introduces artificial downward bias but dramatically minimizes variance by shrinking extreme outlier over-estimates. This makes it an ideal fit for machine learning algorithms, signal filtering, and financial asset management.

Average Variance vs. MSE

The core of this evaluation lies in the definition of Mean Squared Error: $MSE = Bias^2 + Variance$. An estimator can be perfectly unbiased but still be a dangerous choice if its predictions swing wildly between extremes. This instability is exactly why $n - 1$ fails to minimize total error.

By dividing by a small number like $n - 1$ when the sample size is small, any outlier distance gets amplified exponentially. This dramatically widens the variance of the estimator itself. Conversely, dividing by $n + 1$ deliberately introduces negative bias but acts as a regularizer, compressing the entire distribution curve. The resulting reduction in variance far outweighs the bias penalty, making it highly probable that any single sample variance calculation will fall closer to the true population value.

Where the Bessel's Correction Factor Shines

Aerospace Engineering and Structural Fatigue

In aircraft manufacturing, small material batches are tested for stress limits. If engineers use the minimum MSE divisor ($n + 1$), the systematic underestimation of variance will report an artificially narrow spread of material strengths. If a batch contains a high-variance structural flaw, the $n + 1$ divisor will mask the weakness, leading to early component fatigue and catastrophic structural failure mid-flight.

Biomedical Diagnostics and Pharmaceutical Dosages

Consider clinical trials for a high-toxicity narrow-therapeutic-index drug. If sample variance in drug clearance rates is calculated using $n + 1$, the spread is underestimated. The protocol will assume patient responses are tightly clustered, leading to uniform dosing guidelines. In reality, a wider unmapped variance means a significant sub-population will clear the drug too slowly, resulting in toxic over-concentration and preventable patient deaths.

Appendix

Analytical Derivation of Bessel's Correction

Below was rendered in Latex to display the formulas more easily.

Let $X_1, X_2, \dots, X_n$ be independent random variables sampled from a population with true mean μ and true variance σ^2 . The sample mean $\bar{X}$ is defined as:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

We begin by expanding the sum of squared deviations from the sample mean. We subtract and add the population mean μ inside the expression:

$$\begin{aligned} \sum_{i=1}^n (X_i - \bar{X})^2 &= \sum_{i=1}^n [(X_i - \mu) - (\bar{X} - \mu)]^2 \\ &= \sum_{i=1}^n [(X_i - \mu)^2 - 2(X_i - \mu)(\bar{X} - \mu) + (\bar{X} - \mu)^2] \end{aligned}$$

Distributing the summation across the terms yields:

$$\sum_{i=1}^n (X_i - \bar{X})^2 = \sum_{i=1}^n (X_i - \mu)^2 - 2(\bar{X} - \mu) \sum_{i=1}^n (X_i - \mu) + n(\bar{X} - \mu)^2$$

Since $\sum_{i=1}^n (X_i - \mu) = n(\bar{X} - \mu)$, we substitute this into the middle term:

$$\begin{aligned} \sum_{i=1}^n (X_i - \bar{X})^2 &= \sum_{i=1}^n (X_i - \mu)^2 - 2n(\bar{X} - \mu)^2 + n(\bar{X} - \mu)^2 \\ &= \sum_{i=1}^n (X_i - \mu)^2 - n(\bar{X} - \mu)^2 \end{aligned}$$

Next, we apply the expectation operator $E[\cdot]$ to both sides of the equation:

$$E \left[\sum_{i=1}^n (X_i - \bar{X})^2 \right] = \sum_{i=1}^n E [(X_i - \mu)^2] - n \cdot E [(\bar{X} - \mu)^2]$$

By definition, the population variance is $E[(X_i - \mu)^2] = \sigma^2$. Therefore, the first term simplifies to:

$$\sum_{i=1}^n \sigma^2 = n\sigma^2$$

The second expectation term represents the variance of the sample mean itself, which is known to be:

$$E[(\bar{X} - \mu)^2] = \text{Var}(\bar{X}) = \frac{\sigma^2}{n}$$

Substituting these variance values back into our expectation equation gives:

$$\begin{aligned} E\left[\sum_{i=1}^n (X_i - \bar{X})^2\right] &= n\sigma^2 - n\left(\frac{\sigma^2}{n}\right) \\ &= n\sigma^2 - \sigma^2 \\ &= (n-1)\sigma^2 \end{aligned}$$

This proves that the expected value of the uncorrected sum of squared deviations is $(n-1)\sigma^2$. To construct an unbiased estimator S^2 where $E[S^2] = \sigma^2$, we must divide by $n-1$:

$$E[S^2] = E\left[\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2\right] = \frac{1}{n-1} \cdot (n-1)\sigma^2 = \sigma^2$$

Simulation Figures

Figure 1a. Estimator Comparison - Samples Drawn from Normal Distribution

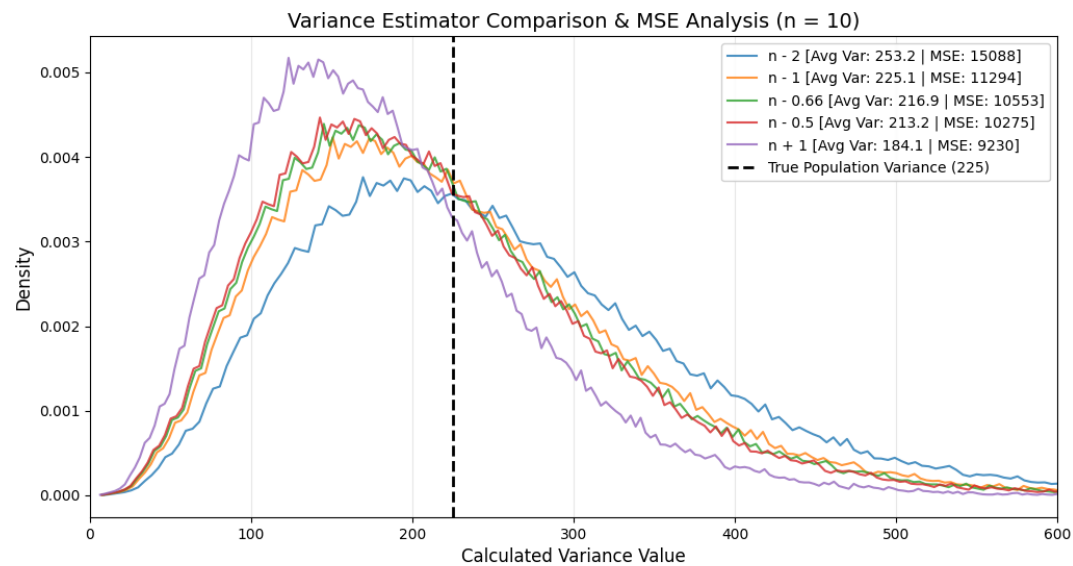


Figure 1b. Estimator Convergence - Samples Drawn from Normal Distribution

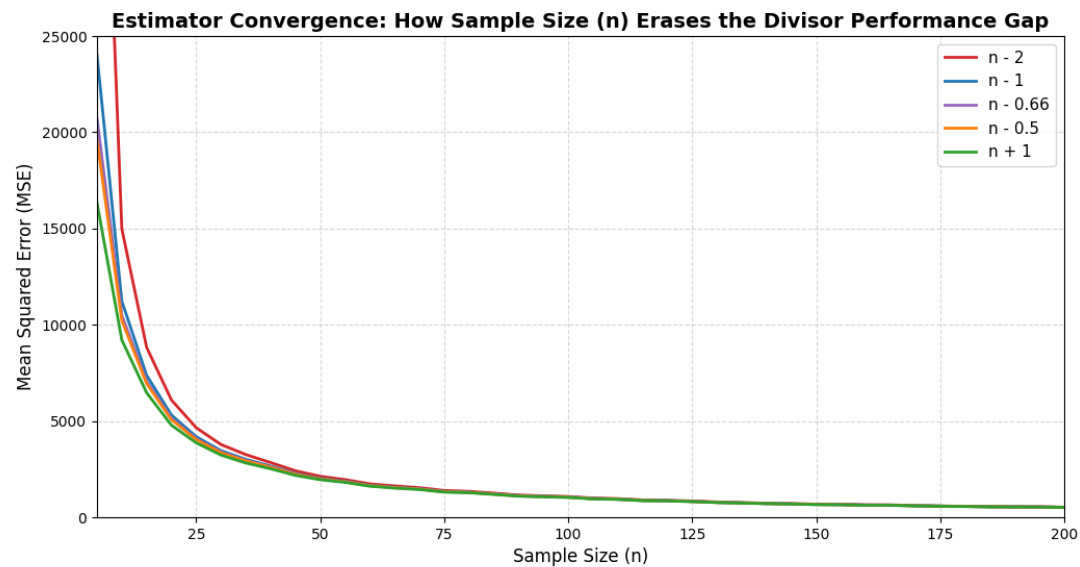


Figure 2a. Estimator Comparison - Samples Drawn from Lognormal Distribution

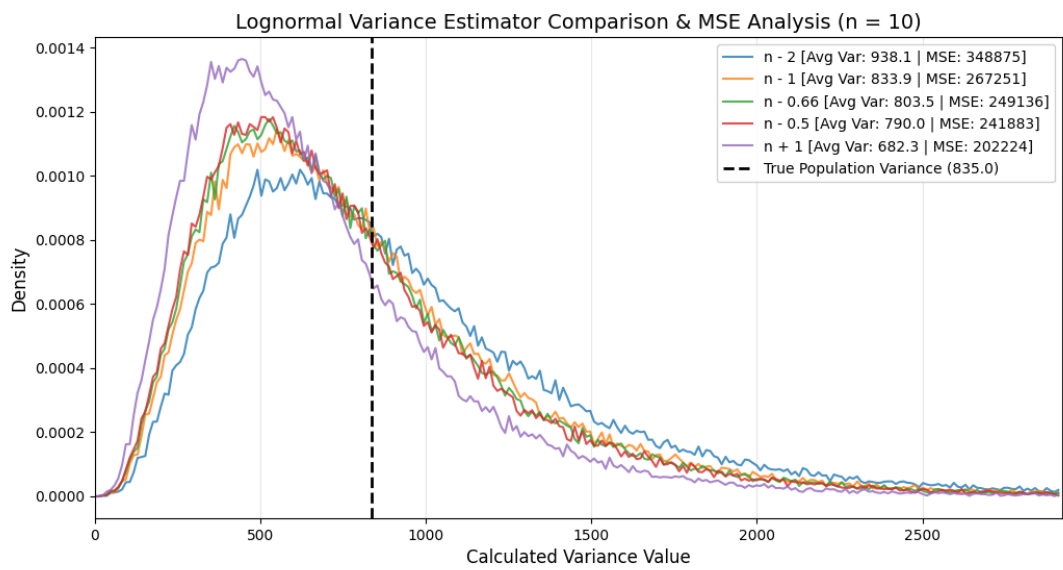
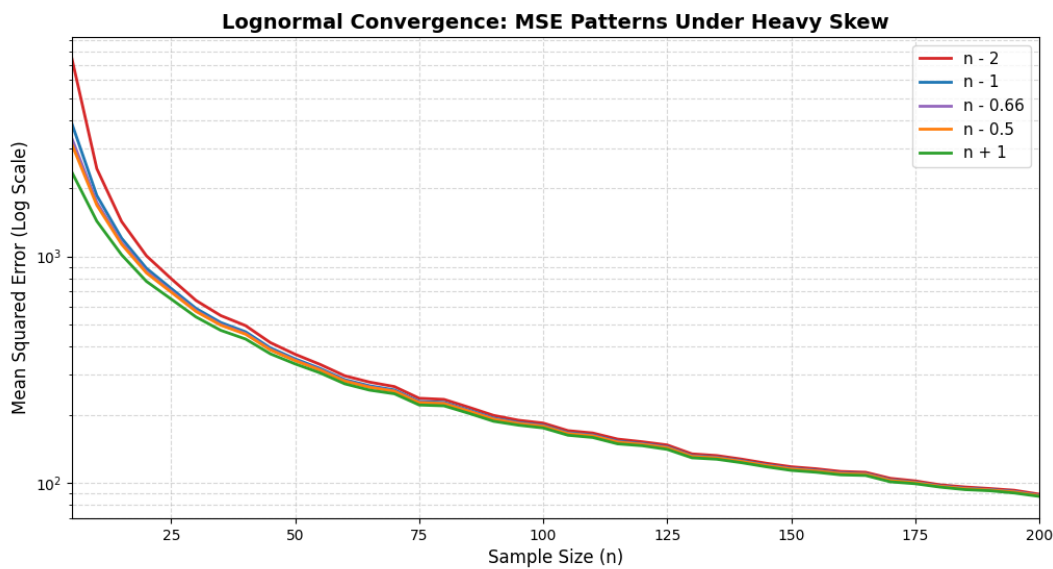


Figure 2b. Estimator Convergence - Samples Drawn from Lognormal Distribution



MSE Simulation Code

```
import numpy as np
import matplotlib.pyplot as plt

# (True Population Variance = 225)
pop_mean, pop_std = 50, 15
true_variance = pop_std**2

n = 10
num_simulations = 100000

samples = np.random.normal(pop_mean, pop_std, size=(num_simulations,
n))
sample_means = np.mean(samples, axis=1, keepdims=True)
sum_squared_deviations = np.sum((samples - sample_means) ** 2,
axis=1)

divisors = {
    "n - 2": n - 2,
    "n - 1": n - 1,
    "n - 0.66": n - 0.66,
    "n - 0.5": n - 0.5,
    "n + 1": n + 1,
}

plt.figure(figsize=(12, 6))

for label, divisor in divisors.items():
    calc_vars = sum_squared_deviations / divisor

    # calculate mean and MSE
    mean_estimate = np.mean(calc_vars)
    mse = np.mean((calc_vars - true_variance) ** 2)

    # plot probability distribution curve
    counts, bins = np.histogram(calc_vars, bins=250, density=True)
    plt.plot(
```

```

        bins[:-1],
        counts,
        label=f"{label} [Avg Var: {mean_estimate:.1f} | MSE:
{mse:.0f}]",
        alpha=0.8,
    )

plt.axvline(
    true_variance,
    color="black",
    linestyle="--",
    linewidth=2,
    label=f"True Population Variance ({true_variance})",
)

plt.title(
    f"Variance Estimator Comparison & MSE Analysis (n = {n})",
    fontsize=14
)
plt.xlabel("Calculated Variance Value", fontsize=12)
plt.ylabel("Density", fontsize=12)
plt.xlim(0, 600)

# Position the legend explicitly in the top right corner
plt.legend(loc="upper right", fontsize=10, framealpha=0.9)
plt.grid(axis="x", alpha=0.3)
plt.show()

```

Tabular Comparison of Variance Estimators

Normal Samples (True Population Variance 225.0)

Divisor	Average Sample Variance	Mean Square Error (MSE)
n-2	253.2	15,087.7
n-1	225.1	11,293.5

n-0.66	216.9	10,552.5
n-0.5	213.2	10,275.1
n+1	184.1	9,230.2

Lognormal Samples (True Population Variance 835.0)

Divisor	Average Sample Variance	Mean Square Error (MSE)
n-2	938.1	348,875
n-1	833.9	267,251
n-0.66	803.5	249,136
n-0.5	790.0	241,883
n+1	682.3	202,224