

How to Generate a Normal Distribution in Google Sheets

Authored by
stats writer

January 16, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Generate a Normal Distribution in Google Sheets*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=126362>

Generating a simulated Normal Distribution, often known as the Gaussian distribution or the bell curve, is a fundamental task in statistical modeling and data analysis. In Google Sheets, this can be achieved efficiently using a specific combination of built-in functions: `=NORMINV(RAND(), mean, standard_deviation)`. This powerful formula integrates the random number generation process with the inverse cumulative distribution function to produce values that accurately reflect the desired distribution characteristics. By utilizing the **RAND** function, which generates a pseudo-random value uniformly distributed between 0 and 1, we provide the probability input. The **NORMINV** function then interprets this probability relative to the specified parameters (the population Mean and Standard Deviation) and outputs the corresponding variate along the Normal Distribution curve. The flexibility of this method allows users to simulate diverse datasets simply by manipulating the defined mean and standard deviation, controlling the central tendency and spread of the resulting data points.

The Essential Formula for Normal Distribution Generation

To accurately generate data points that adhere to a specified normal distribution within Google Sheets, a composite function is required. This technique leverages two core functions to perform the transformation from a uniform random variable (probability) to a normally distributed variable (value). The result is a highly effective method for simulation, particularly useful for demonstrating statistical concepts or generating dummy data for testing purposes. It is critical to structure the implementation by first defining the distributional parameters in separate cells, which allows for effortless adjustment of the simulation without modifying the core generation formula.

The standard structure for generating these values relies on the assumption that the probability input (a value between 0 and 1) corresponds to the cumulative probability of the normal curve. The **NORMINV** function, which calculates the inverse of the normal cumulative distribution, takes this probability and returns the associated data point. Because we need a different, statistically independent data point for each cell, we rely on the **RAND** function to supply a continuously changing probability input, ensuring the simulation yields a diverse sample set.

The full formula, as implemented in Google Sheets, must reference the predetermined parameters for the mean and standard deviation. We strongly recommend using **absolute cell references** for these parameters, ensuring that when the formula is copied down a column to generate the sample, the references to the parameters remain fixed. If, for instance, the mean is stored in cell B1 and the standard deviation in cell B2, the formula takes the following generalized form:

To generate a normal distribution value in Google Sheets, you can use the following formula:

=NORMINV(RAND(), \$B\$1, \$B\$2)

This formula assumes that the mean of the normal distribution is specified in cell **B1** and the standard deviation is specified in cell **B2**.

Deciphering the Component Functions

Understanding the role of each function is essential for effective data generation. The entire mechanism relies on the transformation of a uniformly distributed random number into a normally distributed counterpart. This process is mathematically sound and is based on fundamental principles of probability and statistics, making the generated data appropriate for various modeling tasks where normality is assumed.

First, the **RAND()** function is executed. This function, which takes no arguments, is designed to return a random floating-point number greater than or equal to 0 and strictly less than 1. Crucially, the outputs of **RAND()** are uniformly distributed, meaning any number between 0 and 1 has an equal chance of being generated. This output serves as the input probability for the subsequent step, representing the area under the standard normal curve up to a certain point.

Second, the **NORMINV** function takes over. This function is the Inverse Normal Cumulative Distribution Function. It accepts three arguments: the probability (provided by **RAND()**), the mean of the distribution, and the standard deviation. Given the cumulative probability (p), **NORMINV** determines the value (x) such that the probability of a randomly chosen observation being less than or equal to x is equal to p . By feeding it a random probability from 0 to 1, we effectively sample randomly from the entire range of the specified normal distribution.

By chaining these two functions, we ensure that the generated values are concentrated around the defined mean and spread according to the defined standard deviation, perfectly mimicking the behavior of a genuine normal population. It is important to remember that because **RAND()** is a volatile function, the generated values will automatically recalculate every time the sheet is opened or edited, reflecting the continuous nature of random sampling.

Setting Up Parameters and Initial Configuration

Before implementing the formula, proper sheet setup is crucial for efficiency and clarity. Defining the parameters--the mean (μ) and the standard deviation (σ)--in dedicated, easily identifiable cells simplifies the process of testing different distributions. This modular approach allows users to change the characteristics of the simulated data set instantly without altering the lengthy column of generation formulas.

To begin, select two cells, typically in the top row (e.g., B1 and B2), to serve as the control panel for your distribution. Labeling the adjacent cells (A1 and A2) with "Mean" and "Standard Deviation," respectively, enhances the sheet's readability and ensures that other users can easily understand

the purpose of the input values. For our initial demonstration, we will choose a standard normal distribution, which is characterized by a mean of 0 and a standard deviation of 1.

We will choose 0 and 1, respectively, in cells **B1** and **B2** for our initial parameters. This configuration represents the starting point for simulating the data and provides a baseline for understanding how parameter changes affect the resulting sample values. Once these parameters are entered, the system is ready for the formula implementation phase.

	A	B	C	D
1	Mean	0		
2	Standard Dev	1		
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				

Example: Step-by-Step Generation of the Sample

The practical application involves entering the compounded formula into the first cell where the sample data will reside, followed by dragging the formula down to define the desired sample size. This process generates a column of values, each representing an independent draw from the specified normal distribution. This is the core mechanism by which we transform the theoretical distribution into a tangible dataset.

First, we select the starting cell for our generated data, which we will designate as cell **A5** in this example. This placement keeps the generated data separate from the parameter definitions in column B. We then input the formula, making careful use of absolute references (the dollar sign symbol, **\$**) to anchor the mean and standard deviation to cells B1 and B2, respectively. The use of absolute references is non-negotiable here; failing to use them would result in the formula attempting to reference cells B3, B4, etc., as it is dragged down, leading to errors or inaccurate

results.

Next, we'll type the following formula into cell A5 to generate the first value for our normal distribution:

=NORMINV(RAND(), \$B\$1, \$B\$2)

Once this formula is entered into cell A5, we can execute the generation of the full sample. This is accomplished by using the fill handle (the small square at the bottom right corner of the selected cell) and dragging the formula downwards. The length of this drag determines the **sample size** (N) of the simulated distribution. For instance, dragging the formula down 20 rows (from A5 to A24) creates a sample size of 20, providing sufficient data points for preliminary analysis.

We can then click and drag this formula down to 20 more cells to create a normal distribution with a mean of 0, standard deviation of 1, and sample size of 20:

A5 fx =NORMINV(RAND(), \$B\$1, \$B\$2)

	A	B	C
1	Mean	0	
2	Standard Dev	1	
3			
4	Normally Distributed Dataset		
5	0.1450583517		
6	0.134012293		
7	1.038954005		
8	1.164056097		
9	-0.7874814392		
10	-0.9134705547		
11	-0.5036595368		
12	1.1698085		
13	0.9799448661		
14	-0.229996199		
15	-0.7864443291		
16	1.219634297		
17	-0.05671051713		
18	-1.385628584		
19	-1.209754203		
20	-0.9643241959		
21	1.594120502		
22	0.7275952772		
23	-0.3812599504		
24	-1.074318183		
25			

If you change the values for the mean and standard deviation in cells **B1** and **B2**, respectively, then the values in cells **A5:A24** will automatically update.

The Dynamic Nature of Distribution Control

One of the most powerful features of this setup is the immediate responsiveness to changes in the parameters. Because the formulas in column A rely on the absolute references to cells B1 and B2, updating those two control cells instantly recalculates the entire sample data set, transforming the simulated distribution to match the new specifications.

For example, if the requirement shifts from a standard normal distribution (Mean=0, SD=1) to a distribution representative of, say, standardized test scores (where the mean might be 30 and the standard deviation 4), only cells B1 and B2 need to be edited. Upon changing B1 to 30 and B2 to

4, the values throughout the sample column (A5:A24) will automatically adjust. This provides a dynamic, exploratory environment where researchers or students can instantly visualize the effect of changes to the core statistical moments of the distribution.

This dynamic recalculation is not merely a convenience; it is fundamental to conducting simulations and sensitivity analyses. The ability to pivot between different distributions--such as shifting from a narrow distribution (small SD) to a wide distribution (large SD)--allows for rapid prototyping and validation of theoretical models against simulated data. This feature underscores the utility of structuring the parameters separately from the generation mechanism.

For example, suppose we change the mean to 30 and the standard deviation to 4:

	A	B	C
1	Mean	30	
2	Standard Dev	4	
3			
4	Normally Distributed Dataset		
5	26.69937517		
6	40.63808269		
7	31.39525907		
8	33.30603359		
9	32.82801489		
10	27.33711902		
11	36.33608178		
12	30.91943013		
13	28.88518237		
14	14.75534439		
15	23.00705065		
16	26.29440077		
17	22.75157041		
18	31.54584068		
19	27.48490055		
20	29.46100582		
21	37.53087763		
22	33.02440607		
23	27.35387934		
24	30.70172864		
25			

The new values represent a normal distribution with a mean of 30, standard deviation of 4, and

sample size of 20.

Analyzing and Interpreting the Generated Sample

Once the sample data is generated, the next essential step is to analyze and interpret the results to confirm that the simulated sample accurately reflects the theoretical distribution defined by the parameters. Although the generated values are technically pseudo-random, a sufficiently large sample size should exhibit characteristics closely aligned with the input mean and standard deviation.

To verify the simulation, users should calculate the descriptive statistics of the generated data set (A5:A24). Functions like `=AVERAGE(A5:A24)` and `=STDEV.S(A5:A24)` can be employed to calculate the sample mean and sample standard deviation, respectively. Due to the nature of random sampling, these calculated sample statistics will not perfectly match the input population parameters (B1 and B2), but they should be reasonably close. The discrepancy between the sample statistics and the population parameters highlights the concept of **sampling variability**, a key statistical principle.

Furthermore, plotting the generated data using a histogram is highly recommended. In Google Sheets, selecting the data and inserting a chart, then changing the chart type to a histogram, will visually display the distribution's shape. For a successful simulation, the histogram should approximate the expected bell curve shape, centered around the input mean (e.g., 30) and exhibiting spread proportional to the input standard deviation (e.g., 4). Visual confirmation is often the most intuitive way to ensure the simulation is functioning correctly.

Expanding the Scope: Increasing Sample Size and Robustness

The accuracy of statistical simulations is profoundly influenced by the sample size. While a sample size of 20 is suitable for introductory examples, practical applications often require significantly larger datasets to minimize sampling error and achieve a distribution that closely mirrors the theoretical normal curve. Generating hundreds or even thousands of data points enhances the robustness of the simulation.

Increasing the sample size is straightforward: simply drag the formula down to a greater number of rows. For instance, to generate a sample size of 500, the formula must be extended down 500 rows, from A5 through A504. While this expands the dataset, the underlying mechanism remains identical, and the sheet's dynamic recalculation capabilities handle the increased complexity without user intervention. A larger sample size generally results in calculated sample means and standard deviations that are much closer to the input population parameters defined in B1 and B2.

This method of generating data is invaluable for tasks such as Monte Carlo simulations, where

repeated sampling from a distribution is necessary to estimate complex outcomes. By generating large, statistically reliable samples, Google Sheets becomes a powerful, accessible tool for advanced statistical exploration and risk analysis, using nothing more than the core **NORMINV** and **RAND** functionality.

Further Exploration in Google Sheets Statistics

This technique of generating a Normal Distribution provides a foundation for more complex statistical modeling. Once a simulated data set is established, users can proceed to perform various analytical tasks, such as calculating percentiles, determining probabilities, or comparing distributions.

The following tutorials explain how to perform other common tasks in Google Sheets:

Calculating the standard error of the mean for the simulated data.

Creating control charts to monitor the variability of the generated random process.

Using the simulated data to demonstrate the Central Limit Theorem.

Applying other distribution functions, such as **TINV** or **CHIINV**, to generate non-normal data distributions.

Mastering the combination of **NORMINV** and **RAND** opens up substantial possibilities for statistical education and practical data simulation within the accessible environment of Google Sheets.