

# How to Generate a Normal Distribution in Google Sheets

Authored by  
**stats writer**

February 1, 2026

## RECOMMENDED CITATION

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

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128947>

Generating a normal distribution, often referred to as the Gaussian distribution or the bell curve, is a fundamental task in fields ranging from finance and physics to complex statistical calculations. While specialized software is often employed, the power and accessibility of Google Sheets allow users to simulate this crucial statistical phenomenon easily using built-in functions. The core method relies on the `NORMINV` function, which is designed to perform the inverse cumulative distribution function for the normal distribution, effectively turning uniformly random inputs into normally distributed outputs.

This technique provides a straightforward, repeatable process for generating synthetic datasets that mimic real-world phenomena where values tend to cluster around a central mean. Understanding how to implement this effectively is paramount for tasks such as simulating measurement errors, modeling population characteristics, or validating statistical hypotheses. By leveraging the interplay between the `NORMINV` function and a random number generator, we can precisely control the parameters--specifically the mean and the standard deviation--that define the shape and location of the generated bell curve.

The subsequent sections will provide a detailed, step-by-step guide on how to configure your sheet, write the requisite formulas, and generate thousands of normally distributed data points. This powerful capability transforms Google Sheets from a simple spreadsheet application into a viable tool for basic data simulation and sophisticated exploratory data analysis.

## How to Generate a Normal Distribution in Google Sheets

### Understanding the Core Functions: NORMINV and RAND

To successfully generate a normal distribution, it is essential to understand the roles played by the two primary functions utilized in the formula: `NORMINV` and `RAND`. These functions work in tandem to convert a purely random input into a value that conforms to the specified distribution parameters. This method is mathematically sound and commonly used in computational statistics for Monte Carlo simulations.

The `RAND()` function serves as the starting point. It generates a random, floating-point number between 0 and 1 (inclusive of 0, exclusive of 1). Crucially, these numbers are generated from a uniform distribution, meaning every value between 0 and 1 has an equal probability of being selected. When we use this function repeatedly down a column, we create a large sample of uniformly distributed probability values, which act as inputs for the inverse function.

The `NORMINV(probability, mean, standard_dev)` function is the true engine of the simulation. It calculates the inverse of the normal cumulative distribution function. In simpler terms, given a

probability (the random number from `RAND()`), a mean, and a standard deviation, `NORMINV` returns the value from the distribution that corresponds to that cumulative probability. For instance, if you input a probability of 0.5, `NORMINV` returns the value equal to the mean, as 50% of the distribution lies below the mean.

## The Definitive Formula for Generating Data

The combination of these two functions creates the powerful statistical tool required for the simulation. By nesting `RAND()` inside `NORMINV`, we ensure that the input probability changes every time the sheet recalculates, thereby generating unique, normally distributed data points. The formula structure is optimized for efficient use across a large number of cells, allowing you to control the simulation parameters from designated input cells.

To generate the necessary data points in Google Sheets, you should use the following structured formula. Note the use of absolute references (the dollar signs) for the parameter cells. This critical formatting ensures that when you drag the formula down to populate hundreds of rows, the reference to the parameters (mean and standard deviation) remains constant, while the `RAND()` function changes in every single row.

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

This formula assumes that the desired mean ( $\mu$ ) of the normal distribution is specified in cell **B1** and the desired standard deviation ( $\sigma$ ) is specified in cell **B2**. The resulting data set, generated by copying this formula down, will exhibit the statistical properties defined by these parameters.

## Setting Up Your Simulation Parameters

Before entering the formula, it is essential to organize your spreadsheet by dedicating specific cells to the parameters that define your normal distribution. This methodology provides a centralized control panel for your simulation, allowing for instantaneous changes and easy experimentation without modifying the core generation formula itself. We recommend placing these parameters at the top of the sheet and using them as absolute references, as demonstrated in the previous section.

For clarity and ease of use, label cells in column A (e.g., A1 and A2) to indicate what the adjacent cells in column B represent. Cell **B1** should be designated for the mean (the central location of the distribution), and cell **B2** should contain the standard deviation (a measure of the dispersion or spread of the data). A larger standard deviation results in a wider, flatter curve, while a smaller standard deviation yields a taller, narrower curve.

Once the parameters are set, you can then proceed to implement the formula starting in a

designated data column (e.g., cell A5). The key advantage of this setup is the dynamic nature of Google Sheets: if you change the mean or standard deviation in cells B1 or B2, the entire generated dataset in column A will automatically recalculate and update, reflecting the new distributional characteristics.

## Step-by-Step Example: Generating a Standard Normal Distribution

To illustrate this technique, we will generate data that follows a standard normal distribution, which is characterized by a mean of 0 and a standard deviation of 1. This particular distribution is commonly used as a baseline reference point in sophisticated statistical calculations and modeling.

We begin by inputting the parameters into the designated control cells. For this example, we populate the sheet as follows, establishing the fundamental characteristics of our simulated data set:

In cell **B1**, enter the value 0 (representing the Mean).

In cell **B2**, enter the value 1 (representing the Standard Deviation).

This configuration prepares the sheet for the data generation step, where the formula will reference these specific values:

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

Next, we input the generation formula into the first data cell. We will use cell A5 as our starting point. Type the complete formula, ensuring that the absolute references to B1 and B2 are correctly

included, guaranteeing continuity as the formula is replicated across the column:

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

Once the formula is entered in A5, the cell will display the first randomly generated value from the specified normal distribution. To generate a robust sample size, click on the bottom-right corner of cell A5 and drag the formula down to encompass the desired number of data points. For instance, dragging the formula down to cell A24 generates a sample size of 20, providing a small but illustrative example of the generated data:

	A	B	C
1	<b>Mean</b>	0	
2	<b>Standard Dev</b>	1	
3			
4	<b>Normally Distributed Dataset</b>		
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			

## Scaling and Replicating the Simulation

While the example above uses a modest sample size of 20 data points, the real utility of this method comes from scaling the simulation to thousands of data points. A larger sample size is generally required to accurately represent the true shape of the theoretical normal distribution, making the resulting empirical data more statistically reliable for subsequent analysis or modeling.

To expand the simulation, simply continue dragging the formula down column A. For high-fidelity simulations, generating 1,000 to 10,000 data points is advisable. Remember that because the `RAND()` function is volatile, all values in cells **A5:A** will recalculate every time the spreadsheet is opened, edited, or updated. If you need a static dataset for analysis, it is necessary to copy the generated values and paste them back as static values using the "Paste special -> Values only" option.

### Adjusting Parameters: Mean and Standard Deviation

One of the most powerful aspects of using cell references for the parameters is the ability to instantly visualize the effect of changing the mean and standard deviation. Since the formula references cells **B1** and **B2** absolutely, modifying these input values causes the entire generated dataset (cells **A5:A24** in our example) to recalculate immediately according to the new distribution characteristics.

For example, suppose an analyst needs to simulate test scores with an expected average of 30 and a typical variability of 4. We simply update our control panel: change the mean in cell **B1** to 30 and the standard deviation in cell **B2** to 4. All values in the generated column will instantly shift to reflect this new distribution centered at 30, while maintaining a standardized spread defined by 4. This change is demonstrated below:

The new values represent a normal distribution with a mean of 30, standard deviation of 4, and sample size of 20. Notice how the values, which previously centered around 0, are now centered around 30, confirming the successful dynamic update of the simulation based on the revised parameters:

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

	A	B	C
1	<b>Mean</b>	30	
2	<b>Standard Dev</b>	4	
3			
4	<b>Normally Distributed Dataset</b>		
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			

## Visualizing the Data: Creating a Histogram

Generating the raw numbers is only the first step; to confirm that the output truly follows a normal distribution, it is essential to visualize the data using a histogram. A histogram groups data points into bins and displays the frequency of each bin, allowing the user to visually inspect the resulting distribution shape. For a large sample generated using the `NORMINV` method, the histogram should closely approximate the recognizable bell curve shape.

To create a histogram in Google Sheets, highlight the entire column of generated data (e.g., A5 to A1000). Navigate to the "Insert" menu, and select "Chart." Google Sheets will often automatically suggest a histogram if the data is numerical. If not, select "Chart type" and choose "Histogram

chart." Adjust the bin size in the customization settings to fine-tune the visualization. If the resulting chart shows the familiar symmetrical bell shape, the data generation was successful, validating the use of the `NORMINV` and `RAND()` functions for this purpose.

## Practical Applications in Data Analysis

The ability to generate normally distributed data directly within a spreadsheet environment like [Google Sheets](#) opens up numerous possibilities for practical applications in education, research, and business analysis. This capability is frequently employed in introductory statistics courses to demonstrate complex concepts or in business modeling for sensitivity analysis.

Key applications include:

**Monte Carlo Simulations:** Generating large sets of random, normally distributed numbers is the foundation of many Monte Carlo methods used to model uncertainty and predict potential outcomes in complex financial models or project risk assessments.

**Hypothesis Testing Validation:** Researchers often need synthetic datasets with known properties to test the robustness and accuracy of new statistical tests or algorithms before applying them to real-world, messy data.

**Teaching Statistical Concepts:** Educators can use this function to quickly create visual examples of the Central Limit Theorem or demonstrate how changes in the standard deviation affect the spread of a population, significantly aiding in the comprehension of core statistical concepts.

Mastering this technique is a powerful step toward performing more advanced statistical calculations and data modeling directly within the accessible ecosystem of [Google Sheets](#).

## Additional Google Sheets Tutorials

The following tutorials explain how to perform other common tasks in Google Sheets, complementing your new ability to generate custom distributions: