

How to Generate a Random Sample in Google Sheets: A Step-by-Step Guide

Authored by
stats writer

December 3, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Generate a Random Sample in Google Sheets: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=104257>

Selecting a truly random sample from a large dataset is a fundamental requirement in statistical analysis, quality control, and testing methodologies. Whether you are conducting A/B testing or need a representative subset of data for quick processing, the ability to generate a reliable, unbiased sample is essential. Fortunately, Google Sheets provides a straightforward yet powerful mechanism for achieving this using its built-in mathematical functions.

This comprehensive tutorial outlines the exact, five-step process required to select a fixed-size random subset of rows from any given data range in Google Sheets. We rely primarily on the volatile function **RAND()**, which serves as a powerful tool for assigning a unique, temporary sorting key to each row of your data. This technique ensures that every observation has an equal probability of being selected, thus meeting the criteria for a simple random sample.

Understanding the RAND() Function and Volatility

The core of this random selection technique lies in the **RAND()** function. This function generates a pseudorandom floating-point number between 0 (inclusive) and 1 (exclusive). When used repeatedly across a column, it assigns a unique, random numerical identifier to every corresponding row in your dataset. This number is then used as a sorting index.

It is critical to understand that **RAND()** is a volatile function. A volatile function recalculates its output every time a change is made to the spreadsheet, or sometimes even when the sheet is opened. While this volatility is crucial for generating the initial randomization, it also presents a significant challenge, as the assigned random numbers will constantly shift unless they are converted into static values. The stabilization process detailed below specifically addresses how to fix these random values to ensure a consistent sampling result before extraction.

Step 1: Preparing the Dataset for Sampling

Before initiating the sampling process, your data must be properly structured. Ensure that the dataset you intend to sample is contained within contiguous cells, typically organized with headers in the first row and data points occupying subsequent rows. For this demonstration, we will assume a basic single-column dataset, although this method scales perfectly for tables containing many columns (e.g., customer records, transaction logs, etc.).

As the first action, enter or confirm your data entries within Column A. For illustrative purposes, we will use a small list of numerical values ranging from 1 to 20 across multiple rows. This forms the basis of our primary data pool from which the random sample will be drawn. Ensure that the column immediately adjacent to your data (Column B in this case) is empty, as this is where we will insert our random sorting keys.

For example, our initial setup might look like this:

	A	B	C	D	E	
1	Raw Data					
2	5					
3	6					
4	6					
5	8					
6	10					
7	12					
8	13					
9	13					
10	14					
11	15					
12	17					
13	20					
14	21					
15	22					
16	27					
17						
18						
19						

Step 2: Generating the Temporary Random Key List

The next critical step involves using the **RAND()** function to associate a random numerical key with every row in your data. This key is what will determine the ultimate randomized order of the observations during the sorting phase.

Start by navigating to the cell adjacent to the first data point. If your data starts in A2, you will input the formula into cell B2. Type the following formula:

```
=RAND()
```

Upon pressing Enter, B2 will immediately display a random decimal value between 0 and 1. Remember, this value is currently volatile, meaning it is subject to constant change.

To apply this random key to all data points, click on cell B2 and use the fill handle (the small square at the bottom right corner of the cell) to drag the formula down, covering all corresponding rows in Column A. Alternatively, you can copy the formula and paste it down the entire column. This action populates Column B with unique, temporary random values, one for each data entry.

The sheet should now resemble this visualization:

	A	B	C	D	E
1	Raw Data	Random Number			
2	5	0.09745187759			
3	6	0.6076916209			
4	6	0.5261692792			
5	8	0.4387673992			
6	10	0.6933042649			
7	12	0.6958931911			
8	13	0.9804370318			
9	13	0.3575659463			
10	14	0.2300584962			
11	15	0.6583574369			
12	17	0.7148310576			
13	20	0.1904448262			
14	21	0.9721330673			
15	22	0.8297472637			
16	27	0.8078631107			
17					
18					
19					
20					
21					

Step 3: Stabilizing the Random Values (Paste Special)

As emphasized earlier, the values in Column B are currently formulas, not static numbers, meaning they will change the moment the spreadsheet recalculates or when any cell is edited. To successfully execute the random sample selection, we must immediately convert these formulas into their current numerical results--a process known as "fixing" or "stabilizing" the values.

Begin by selecting all the cells in Column B that contain the **RAND()** formula (e.g., B2 through B16). Copy these values using the keyboard shortcut **Ctrl + C** (or **Cmd + C** on Mac). This action copies the underlying formulas and their instantaneous numerical results.

Crucially, do not paste these values back into Column B immediately. Instead, click on an empty cell in an adjacent column, such as C2. Right-click on C2, navigate to the **Paste special** menu option, and select **Paste values only**. This operation pastes the numerical results of the random formulas, eliminating the volatility. Note that the values in Column B may instantly shift upon this action, but this is irrelevant, as we have secured the necessary fixed values in Column C.

The stabilization process generates a static key list, ready for the sorting phase:

	A	B	C	D	E
1	Raw Data	Random Number			
2	5	0.5565240853	0.09745187759		
3	6	0.07579988487	0.6076916209		
4	6	0.6808403835	0.5261692792		
5	8	0.9249396857	0.4387673992		
6	10	0.4131714107	0.6933042649		
7	12	0.9672858855	0.6958931911		
8	13	0.3586177816	0.9804370318		
9	13	0.4785418919	0.3575659463		
10	14	0.6651376263	0.2300584962		
11	15	0.6641707109	0.6583574369		
12	17	0.2221696466	0.7148310576		
13	20	0.5734229424	0.1904448262		
14	21	0.863426505	0.9721330673		
15	22	0.971635555	0.8297472637		
16	27	0.9551285087	0.8078631107		
17					
18					
19					
20					
21					

The final part of this step involves replacing the volatile formulas in Column B with the stabilized numbers from Column C. Highlight the static values in Column C, copy them, and then paste them back into the corresponding cells in Column B (using **Paste values only** again is recommended to maintain clean data). Once confirmed, you may delete the contents of Column C, as Column B now holds the permanent, fixed, pseudorandom sorting keys.

This replacement ensures that Column B acts as the definitive index for the randomization process:

	A	B	C	D	E
B2		<i>fx</i> 0.0974518775946126			
1	Raw Data	Random Number			
2	5	0.09745187759			
3	6	0.6076916209			
4	6	0.5261692792			
5	8	0.4387673992			
6	10	0.6933042649			
7	12	0.6958931911			
8	13	0.9804370318			
9	13	0.3575659463			
10	14	0.2300584962			
11	15	0.6583574369			
12	17	0.7148310576			
13	20	0.1904448262			
14	21	0.9721330673			
15	22	0.8297472637			
16	27	0.8078631107			
17					
18					
19					
20					

Step 4: Sorting the Dataset Using the Random Keys

With the permanent random sorting keys in place, the next step is to randomize the order of the original dataset. This is achieved through a standard sort operation that uses the random key column (Column B) as the primary sorting criterion.

Select the entire range of data that needs to be randomized, ensuring you include both the original data column (Column A) and the new random key column (Column B). For our example, this range is A2:B16. It is crucial to select both columns simultaneously so that the original data values remain physically linked to their respective random keys during the sort.

Once the range is selected, navigate to the **Data** tab located in the top ribbon menu of Google Sheets. Within the Data menu, choose the **Sort range** option. This action opens a dialog box prompting you to define the sorting parameters.

	A	B	C	D	E	F	G
1	Raw Data	Random Number					
2	5	0.09745187759					
3	6	0.6076916209					
4	6	0.5261692792					
5	8	0.4387673992					
6	10	0.6933042649					
7	12	0.6958931911					
8	13	0.9804370318					
9	13	0.3575659463					
10	14	0.2300584962					
11	15	0.6583574369					
12	17	0.7148310576					
13	20	0.1904448262					
14	21	0.9721330673					
15	22	0.8297472637					
16	27	0.8078631107					
17							
18							
19							
20							
21							
22							

In the Sort range dialog box, specify the following criteria:

Ensure the "Data has header row" checkbox is appropriately set based on your selection.

For the primary sort column, select the column corresponding to the random keys (Column B).

Set the sorting order to either **A → Z** (Ascending) or **Z → A** (Descending). Since the keys are derived from a uniform distribution (0 to 1), the sorting direction does not impact the randomness of the final sequence, but simply ensures a systematic arrangement.

Click the **Sort** button. The sheet will instantly reorder all rows based on the random keys in Column B, effectively scrambling the original data sequence. This randomized arrangement ensures that the first 'n' rows, regardless of their previous position, constitute a statistically unbiased simple random sample.

The resulting sheet will now show the data sorted by the generated random number:

A2:B16 fx 5

	A	B	C	D	E
1	Raw Data	Random Number			
2	5	0.09745187759			
3	20	0.1904448262			
4	14	0.2300584962			
5	13	0.3575659463			
6	8	0.4387673992			
7	6	0.5261692792			
8	6	0.6076916209			
9	15	0.6583574369			
10	10	0.6933042649			
11	12	0.6958931911			
12	17	0.7148310576			
13	27	0.8078631107			
14	22	0.8297472637			
15	21	0.9721330673			
16	13	0.9804370318			
17					
18					
19					
20					
21					

Step 5: Extracting the Final Random Sample

The most challenging steps--generating and stabilizing the random keys, and then sorting the data--are now complete. The dataset is fully randomized, meaning the observations are listed in an order that is statistically independent of their initial arrangement or underlying value. The data in Column A is now randomly ordered based on the indices created in Column B.

The final task is to define and extract your desired subset size. If you require a random sample of size 5 ($n=5$), you simply select the first five rows of data from your now-randomized dataset (i.e., cells A2 through A6). These rows represent the observations selected by chance.

Highlight the required number of rows in the data column (Column A). These highlighted rows represent your simple random sample, ready for further analysis or export. You may now delete the temporary random key column (Column B) if it is no longer needed.

Visualizing the selection of the first five rows:

	A	B	C	D	E
1	Raw Data	Random Number			
2	5	0.09745187759			
3	20	0.1904448262			
4	14	0.2300584962			
5	13	0.3575659463			
6	8	0.4387673992			
7	6	0.5261692792			
8	6	0.6076916209			
9	15	0.6583574369			
10	10	0.6933042649			
11	12	0.6958931911			
12	17	0.7148310576			
13	27	0.8078631107			
14	22	0.8297472637			
15	21	0.9721330673			
16	13	0.9804370318			
17					
18					
19					
20					

In this specific example resulting from the sort operation, the extracted random sample (n=5) would comprise the values: **5, 20, 14, 13, and 8**. This list constitutes a valid, unbiased random subset of the original data.

Conclusion and Advanced Sampling Considerations

The method utilizing the **RAND()** function and the stabilization technique provides an efficient and reliable way to generate a simple random sample in Google Sheets. This approach is highly recommended for its precision and ease of implementation, ensuring that the selection process is free from selection bias.

While this technique covers simple random sampling, more complex statistical needs, such as stratified sampling or weighted sampling, may require advanced functions like **QUERY** combined with filters or array formulas. However, for the majority of users requiring a quick, statistically sound random selection from a dataset, the sorting method presented here remains the optimal choice.

For those utilizing other platforms, similar logical steps--generating a random key, stabilizing it, and then sorting--are often employed. The following resources explain analogous procedures in other statistical software:

How to Select a Random Sample in R

Using Microsoft Excel for Random Data Selection

Implementing Sampling Procedures in Python (e.g., using the Pandas library)

ARABPSYCHOLOGY.COM