How do I select a random sample in Excel?

Authored by

stats writer

December 18, 2025

Selecting a truly random sample from a large population or dataset is a fundamental requirement in statistical analysis, quality control, and auditing. Microsoft Excel offers several robust methods to achieve this goal, ensuring unbiased selection for your research or reporting needs. While alternative approaches exist, such as using the **RANDBETWEEN** function or dedicated statistical tools like the Data Analysis ToolPak's Random Number Generation feature, the most straightforward and flexible method involves leveraging Excel's inherent mathematical capabilities, specifically the RAND() function.

This comprehensive guide details a step-by-step process using the **RAND() function** to efficiently shuffle and select a desired subset from any given range of data. This powerful technique transforms your structured data into a randomly ordered list, allowing you to easily pick the top 'n' records as your representative sample. We will also explore the critical step of converting volatile formulas into static values, which is essential for preserving the integrity and accuracy of your final random selection.

1. Understanding Random Sampling in Excel

The requirement to extract a representative subset, or random sample, is paramount when dealing with large volumes of data where analyzing the entire population is time-consuming or computationally impractical. By using a properly selected sample, analysts can draw statistically reliable conclusions about the entire dataset without exhaustive effort. Excel provides a highly accessible environment for performing this type of operation, especially since its core functions are designed for high-speed calculation and data manipulation.

When discussing randomness in a computational context like Excel, we are generally relying on pseudorandom number generation. The values returned by functions like **RAND()** are not truly random but are calculated using sophisticated algorithms based on an initial seed value. However, for practical applications such as data sampling in statistics or auditing, this level of randomness is more than adequate to ensure that every row in your original dataset has an equal, unbiased probability of being selected.

The methodology described in this tutorial hinges on the principle of pairing each data record with a unique, randomly generated decimal number. This number serves as a temporary sort key. Once the entire dataset is sorted based on these arbitrary keys, the original order is completely scrambled, creating a statistical shuffle. By choosing the first few rows after this comprehensive shuffle, we effectively isolate a statistically sound random sample.

2. The Core Tool: Leveraging the RAND() Function

The **RAND() function** is the technological linchpin of this entire data sampling technique. It is an

extremely straightforward function that requires no arguments, structured only as `=RAND()`. When entered into a cell, it instantaneously returns a decimal value that falls within the range [0, 1), meaning it is greater than or equal to 0 and strictly less than 1. This continuous, uniform distribution of numbers provides the perfect basis for achieving a completely unbiased shuffle.

A vital characteristic of **RAND()**, which must be managed carefully, is its volatility. A volatile function recalculates its output every time a change is made anywhere in the workbook--including data entry, deletion, or even external links refreshing. If we were to apply **RAND()** and then attempt to sort the data, the random keys would immediately recalculate upon completion of the sort operation, potentially shifting the sample boundaries before final selection. This inherent instability mandates a critical procedural step: converting the volatile formulas into static, fixed values, a process detailed in Step 3.

While other random number functions are available, such as **RANDBETWEEN(bottom, top)**, which generates a random integer within a specified range, **RAND()** is generally the preferred choice for statistical sampling. Using decimals minimizes the chance of generating identical sort keys for multiple records, thus eliminating potential ties that could introduce a subtle bias during the sorting process of large populations. The uniform distribution provided by RAND() function ensures maximum randomness.

3. Step 1: Preparing Your Source Dataset

Before any randomization can occur, it is essential to ensure that your source data is correctly isolated and structured. All values or records that constitute your potential population must reside within continuous rows and columns in a singular table format. This careful preparation guarantees that the subsequent steps of applying the random function and sorting affect all records simultaneously and equally, preserving the integrity of each record.

For the purpose of this practical illustration, we will utilize a basic dataset consisting of 20 distinct numerical values entered into a single vertical range, specifically Column A. These 20 values collectively represent the entire population from which our sample will be drawn. It is important to note that while our example uses simple numbers, this randomization procedure is equally effective for complex records involving multiple columns of text, dates, or financial metrics, provided the entire record is selected and treated as a single unit during the sort.

It is best practice to verify the precise boundaries of your data range and ensure that a header row, if present, is clearly distinct from the data itself. If a header exists, it should be excluded from the random key generation step. In our simple setup, we use cells A2 through A21 to house our 20 population values, ensuring clear delineation for the process to follow:

	A	B	C	D	E
1	**Raw Data**				
2	8				
3	29				
4	12				
5	4				
6	17				
7	24				
8	24				
9	22				
10	27				
11	18				
12	34				
13	13				
14	31				
15	24				
16	26				
17	5				
18	20				
19	16				
20	16				
21	7				
22					
23					
24					
25					

4. Step 2: Generating Temporary Random Sort Keys

The next indispensable phase involves creating the temporary keys that Excel will use to shuffle the data. These keys must be generated in a column immediately adjacent to the dataset to ensure they remain structurally linked during the sorting operation. This pairing is crucial for maintaining data integrity.

Move to the cell corresponding to the first data entry in your dataset (in our example, this is cell B2). In this cell, input the formula `=RAND()`. Upon execution, cell B2 will display a unique decimal value between 0 and 1. This value is the initial, temporary sorting key assigned to the corresponding data point in A2.

To propagate this volatile formula across the entire range of your population, employ the quick fill handle feature. Locate the small square at the bottom right corner of cell B2--this is the fill handle. Hover your cursor over this handle until it transforms into a thin black **+** sign, and then double-click. This action swiftly instructs Excel to copy the **=RAND()** formula down the column, precisely matching the extent of the adjacent data population in Column A. This process efficiently assigns a newly generated random number to every single record in your population:

	A	B	C	D
1	**Raw Data**	**Random Number**		
2	8	0.043921669		
3	29	0.68736765		
4	12	0.259600296		
5	4	0.806563658		
6	17	0.290029988		
7	24	0.597754243		
8	24	0.136941987		
9	22	0.213770739		
10	27	0.591114597		
11	18	0.050336217		
12	34	0.713788936		
13	13	0.811501071		
14	31	0.957883244		
15	24	0.1684599		
16	26	0.115222534		
17	5	0.739054968		
18	20	0.423619367		
19	16	0.911546513		
20	16	0.560257811		
21	7	0.878931484		
22				
23				
24				

It is important to notice that at this stage, the values in Column B may constantly change if you perform any action in the spreadsheet. This continuous fluctuation is a definitive confirmation of the function's volatility and underscores why the subsequent conversion step is absolutely vital for freezing the values derived from the RAND() function.

5. Step 3: Fixing the Random Values (Copy & Paste)

Since the values generated by **RAND()** are volatile and recalculate continuously, we cannot use them directly for sorting. Attempting to sort based on a formula-driven column would result in the values shifting during the sort operation itself, which would completely undermine the intended randomization. Therefore, the calculated random values must be permanently converted from dynamic formulas into static numerical data.

Begin by highlighting the entire range of cells containing the **RAND()** formulas in Column B. Use the standard keyboard shortcut **Ctrl + C** to copy the cells and their current calculated outputs to the clipboard. Next, right-click on the first cell of the copied range (cell B2) and access the **Paste Special** menu. From the available options, select the Paste Values icon (represented as a clipboard with '123').

Executing the Paste Values command overwrites the existing formulas with their numeric results. Inspect the cells in Column B; they should now display static decimal numbers, and clicking on any cell in this range should show only the number itself in the formula bar, not the original `=RAND()` formula. This freezing action is depicted below:

	A	B	C	D	E
1	**Raw Data**	**Random Number**			
2	8	0.998997216	0.043922		
3	29	0.024166791	0.687368		
4	12	0.320807811	0.2596		
5	4	0.159879808	0.806564		
6	17	0.627448184	0.29003		
7	24	0.771344909	0.597754		
8	24	0.812447788	0.136942		
9	22	0.521036271	0.213771		
10	27	0.858729193	0.591115		
11	18	0.884189865	0.050336		
12	34	0.463494872	0.713789		
13	13	0.490780896	0.811501		
14	31	0.057939007	0.957883		
15	24	0.127873639	0.16846		
16	26	0.253593241	0.115223		
17	5	0.634975228	0.739055		
18	20	0.142003662	0.423619		
19	16	0.292506681	0.911547		
20	16	0.41637333	0.560258		
21	7	0.53004718	0.878931		
22					
23					
24					
25					

If you opted to paste the static values into a temporary third column (e.g., Column C), you must then highlight the static values and drag or copy them back to replace the volatile formulas in Column B. Excel will prompt you with a dialog box stating: "There's already data here. Do you want to replace it?" Confirm this action by clicking **OK**. Column B is now ready, containing fixed, non-volatile random keys that will not change during the sorting process.

	A	B	C	D	E
1	**Raw Data**	**Random Number**			
2	8	0.043921669			
3	29	0.68736765			
4	12	0.259600296			
5	4	0.806563658			
6	17	0.290029988			
7	24	0.597754243			
8	24	0.136941987			
9	22	0.213770739			
10	27	0.591114597			
11	18	0.050336217			
12	34	0.713788936			
13	13	0.811501071			
14	31	0.957883244			
15	24	0.1684599			
16	26	0.115222534			
17	5	0.739054968			
18	20	0.423619367			
19	16	0.911546513			
20	16	0.560257811			
21	7	0.878931484			
22					
23					
24					

6. Step 4: Sorting and Shuffling the Data

With the permanent random sort keys successfully established in Column B, the dataset is primed for the actual randomization phase. This powerful operation requires selecting both the data column (Column A) and the static random key column (Column B) and applying Excel's native sort function.

Highlight the entire block of data, encompassing both the source data (A2:A21) and the fixed random keys (B2:B21). It is absolutely paramount that the entire block is selected to maintain the established link between each original data point and its unique, assigned random key throughout the sort. Navigate to the **Data** tab on the Excel ribbon and click the **Sort** button found within the Sort & Filter group.

In the subsequent Sort dialog box, clearly specify that the primary sort key is the column containing the random numbers (Column B). The direction of the sort--whether smallest to largest or largest to smallest--is statistically irrelevant to the result, as both directions achieve a complete and thorough randomization of the records in Column A. The sorting algorithm uses the fixed random keys to

arbitrarily reposition the corresponding data records, successfully shuffling the entire dataset into a new, random sequence.

Following the execution of the sort, the original list of values in Column A will be completely disordered. This statistical disorder is the precise outcome required for true random sampling. The values are now ordered based purely on the arbitrarily assigned decimal numbers, achieving an unbiased arrangement that eliminates positional sequence bias:

	A	B	C	D	E
1	**Raw Data**	**Random Number**			
2	8	0.043921669			
3	18	0.050336217			
4	26	0.115222534			
5	24	0.136941987			
6	24	0.1684599			
7	22	0.213770739			
8	12	0.259600296			
9	17	0.290029988			
10	20	0.423619367			
11	16	0.560257811			
12	27	0.591114597			
13	24	0.597754243			
14	29	0.68736765			
15	34	0.713788936			
16	5	0.739054968			
17	4	0.806563658			
18	13	0.811501071			
19	7	0.878931484			
20	16	0.911546513			
21	31	0.957883244			
22					
23					
24					
25					

7. Step 5: Extracting the Final Random Sample

The final step is the most straightforward part of the process: selecting the desired quantity of records from the top of the newly shuffled list. Because the data has been randomized efficiently and without bias, the first 'n' rows now represent a true random sample of size 'n' extracted from the original population.

The determination of the precise sample size ('n') must align with the statistical requirements or auditing scope of your project. If, for instance, you require a random sample of five records, you simply select the first five rows of data from Column A. Using our running example, where our

starting population had 20 values, and we aimed for a 25% sample size, we would select the top five rows from the randomized list.

In this specific illustration, assuming the required sample size is 5, the selected values from the top of the randomized Column A would include 8, 18, 26, 24, and 24. These five values now constitute your representative, randomly selected subset. You are now free to copy and paste these records to a dedicated worksheet for subsequent analysis, discarding the temporary random key column (Column B) if it is no longer required for documentation:

	A	B	C	D	E
1	**Raw Data**	**Random Number**			
2	8	0.043921669			
3	18	0.050336217			
4	26	0.115222534			
5	24	0.136941987			
6	24	0.1684599			
7	22	0.213770739			
8	12	0.259600296			
9	17	0.290029988			
10	20	0.423619367			
11	16	0.560257811			
12	27	0.591114597			
13	24	0.597754243			
14	29	0.68736765			
15	34	0.713788936			
16	5	0.739054968			
17	4	0.806563658			
18	13	0.811501071			
19	7	0.878931484			
20	16	0.911546513			
21	31	0.957883244			
22					
23					

8. Alternative Methods and Advanced Considerations

While the **RAND()** sorting method is highly versatile and generally preferred for its simplicity, it is beneficial to recognize alternative techniques available within Excel for data sampling. These methods often involve either leveraging other mathematical functions or employing advanced statistical add-ins.

For users who possess the **Analysis ToolPak** add-in--which is standard in many Excel installations and accessed via the Data tab--Excel offers a dedicated Sampling tool. This feature allows users to specify the input range and either a precise sample size or a sampling interval. While this dedicated tool can be highly efficient for very large datasets, many practitioners still

favor the **RAND()** method because it does not require activating an external add-in, provides immediate visual confirmation of the data shuffle, and offers greater control over the randomization keys. Regardless of the chosen approach, maintaining an understanding of the concepts behind pseudorandom number generation remains crucial for ensuring the statistical validity of the selected subset.