

How to Rank Values with Ties in Google Sheets

Authored by
stats writer

January 15, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Rank Values with Ties in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=126234>

Analyzing and ranking data is a fundamental requirement across various fields, from academic assessment to complex business metrics. When working with large datasets in tools like Google Sheets, the process of assigning numerical ranks becomes complicated when two or more values are identical--a situation known as a 'tie'. Effectively managing these tied values is crucial for maintaining the integrity and accuracy of your statistical analysis.

Google Sheets offers powerful functions specifically designed to handle ranking, most notably the **RANK** function and its variants. These tools not only assign a rank to every value within a specified range but also employ different methodologies for resolving ties, ensuring flexibility based on the user's analytical needs. Understanding which ranking method to apply--whether assigning the highest rank, calculating an average rank, or employing a tie-breaker--is essential for generating meaningful reports.

This comprehensive guide delves into the three primary strategies available in Google Sheets for ranking values that contain ties. We will meticulously examine the syntax, operational logic, and practical implications of each method, providing you with the expertise necessary to choose the most appropriate approach for your specific data challenges.

The Core Function: Utilizing the Standard RANK Function

The standard **RANK** function is the foundational tool used to determine the position of a specific numerical value within a list of values. Its primary purpose is to assign a rank based on the magnitude of the value relative to others in the defined range, often referred to as the **reference array**. By default, the **RANK** function operates in descending order, meaning the largest value receives the rank of 1, unless the optional `is_ascending` argument is set to 1 (TRUE).

However, the behavior of the standard RANK function when encountering ties is particularly important. When two or more values are exactly equal, this function assigns them the highest available rank in that sequence. For instance, if the top three scores are identical, all three will receive the rank of 1. The next unique score will then skip ranks 2 and 3, and receive a rank of 4. This method is often preferred in competitive scenarios where all top performers deserve the top designation, regardless of how many there are.

The syntax for the basic **RANK** function is straightforward: `=RANK(value, data,)`. Here, `value` is the specific cell you wish to rank (e.g., B2), and `data` is the range of values against which the ranking is performed (e.g., `B2:B11`). Proper use of **absolute references** (e.g., using dollar signs like `B2:B11`) is critical when dragging the formula down a column to ensure the reference range remains fixed.

Method 1: Assigning the Highest Rank to Equal Values

This method utilizes the plain **RANK** function, often chosen when the goal is to identify all data points that achieve a certain maximum threshold. It emphasizes equality among the highest-performing tied values by granting them the same, best possible rank. The subsequent rank is then reserved for the first value that statistically follows the group of ties. This approach ensures that tied entries are treated equivalently in terms of precedence.

Consider a situation where grading criteria mandates that all students achieving the highest possible score must share the first position. If five students score 98%, and this is the highest score, they all receive a rank of 1. The student with the next highest score, say 95%, would then receive a rank of 6 (skipping positions 2, 3, 4, and 5). This mechanism is clear and unambiguous in its designation of top performers.

The implementation requires only the standard **RANK** function, ensuring that the `is_ascending` argument is omitted or set to 0 (for descending order), which is the default behavior suitable for ranking high scores or large values. The formula is structured as follows, targeting cell **B2** within a fixed range:

```
=RANK(B2,$B$2:$B$11)
```

This precise formula will analyze the value in cell **B2** against the entire array from **\$B\$2:\$B\$11** and assign the highest available rank should ties occur. The use of the dollar signs ensures that as this formula is copied down the column, the comparison range remains fixed, providing consistent results across the dataset.

Method 2: Calculating the Average Rank for Tied Values

For more balanced and less aggressive ranking, the **RANK.AVG** function is the preferred choice. Unlike the standard **RANK** function, which always assigns the highest possible rank, **RANK.AVG** calculates the mathematical average of the ranks that the tied values would have individually occupied. This approach is frequently utilized in academic or professional settings where a middle ground is sought to distinguish the clustered performance levels while still acknowledging the tie.

To illustrate, if two values are tied for the 1st and 2nd positions, **RANK.AVG** calculates the average of those two ranks: $(1 + 2) / 2 = 1.5$. Both tied values are then assigned the rank 1.5. This fractional ranking clearly indicates that while the values are superior to the rest of the list, they do not singularly occupy the absolute first position. This method ensures that the sum of all ranks remains consistent, aligning better with certain principles of statistical analysis.

The structure of the **RANK.AVG** function mirrors that of the standard **RANK**, requiring the target

value and the reference range. Because this is a specialized function specifically designed to calculate averages for ties, it provides a cleaner solution than attempting to manually average ranks using multiple formulas. The syntax used for implementing this method is:

```
=RANK.AVG(B2,$B$2:$B$11)
```

By using **RANK.AVG**, you introduce a degree of numerical differentiation without entirely breaking the tie, offering a nuanced view of comparative performance within the **\$B\$2:\$B\$11** range.

Method 3: Sequential Ranking Using a Tie-Breaker Formula

There are instances where simply assigning the same rank (either highest or average) is insufficient. When a strict sequential ordering is required--for example, if you must assign unique ranks 1, 2, 3, etc., even if values are identical--a composite formula leveraging both the RANK function and the COUNTIF function becomes necessary. This sophisticated technique effectively breaks ties based on the physical position of the data point within the spreadsheet array.

The logic behind this tie-breaking formula is to first determine the general rank (using **RANK**) and then adjust that rank downward for any subsequent identical values. The **COUNTIF** function is deployed to count how many times the specific value has appeared already in the preceding rows of the dataset. For the very first instance of a tied value, **COUNTIF** returns 1 (counting the current cell). By subtracting 1 from this count, the first occurrence receives no rank penalty. However, the second occurrence of that value returns a count of 2, leading to a net adjustment of +1 to its rank, thus pushing it to the next sequential position.

This method guarantees that every row receives a unique rank, even if the data value itself is repeated. This is highly useful for processes that demand a definitive ordering, such as assigning prizes or determining priority based on entry sequence. Note the careful implementation of mixed references (e.g., **B\$2:B2**) within the COUNTIF function, which is essential for defining a dynamic, expanding range that checks only the cells above and including the current row.

The complete formula combines these two elements:

```
=RANK(B2,$B$2:$B$11)+COUNTIF(B$2:B2,B2)-1
```

Using this specific structure, the highest rank is assigned to the value that first appears in the dataset, and sequential ranks are assigned to subsequent identical values in the range **B2:B11**.

Practical Application: Setting Up the Ranking Example

To solidify our understanding of these three methodologies, let us examine a practical example

using a typical dataset. Suppose we have recorded the exam scores for ten students in a class. Our objective is to rank these scores from highest (1) to lowest (10), while explicitly demonstrating how each method handles any score ties that occur.

The original dataset, contained within [Google Sheets](#), looks something like the image below, with Student Names in Column A and their respective Exam Scores in Column B:

Suppose we have the following dataset in Excel that shows the exam scores received by various students in some class:

	A	B	C	D
1	Student	Score		
2	Andy	90		
3	Bob	88		
4	Chad	84		
5	Doug	95		
6	Eric	95		
7	Frank	91		
8	Greg	83		
9	Henry	78		
10	Isaac	74		
11	John	85		
12				
13				
14				
15				

We need to create three new columns: one for the Highest Rank (Method 1), one for the Average Rank (Method 2), and one for the Sequential Rank (Method 3). These columns will occupy C, D, and E, respectively, starting from row 2. We are performing a standard descending rank (highest score receives rank 1).

We can type the following formulas into cells **C2**, **D2** and **E2** to apply the different ranking methods, and then drag these formulas down to row 11 to process the entire dataset:

C2: =RANK(B2,\$B\$2:\$B\$11)

D2: =RANK.AVG(B2, \$B\$2:\$B\$11)

E2: =RANK(B2,\$B\$2:\$B\$11)+COUNTIF(B\$2:B2,B2)-1

E2 fx =RANK(B2,\$B\$2:\$B\$11)+COUNTIF(B\$2:B2,B2)-1

	A	B	C	D	E
1	Student	Score	RANK	RANK.AVG	RANK + COUNTIF
2	Andy	90	4	4	4
3	Bob	88	5	5	5
4	Chad	84	7	7	7
5	Doug	95	1	1.5	1
6	Eric	95	1	1.5	2
7	Frank	91	3	3	3
8	Greg	83	8	8	8
9	Henry	78	9	9	9
10	Isaac	74	10	10	10
11	John	85	6	6	6
12					
13					
14					
15					
16					
17					

A careful inspection of the scores reveals a critical tie: **Doug and Eric both received an exam score of 95**, which is the highest score in the dataset. This tie occupies the potential rank positions of 1 and 2. Let's analyze how each method addressed this specific situation, as well as the cascading effect on the subsequent ranks (starting with the score of 90).

Analyzing the Outcomes of Different Tie Handling Methods

Upon applying the three formulas and expanding them to cover all ten students, the resulting sheet clearly illustrates the differences in rank assignment. The visual representation highlights how subtle choices in formula structure can significantly alter the interpretation of the data:

Notice that there are two values that are tied for the highest score: Doug and Eric both received an exam score of 95.

Here is how each ranking method handled this tie:

Method 1: RANK (Highest Rank)

When using the standard **RANK** formula in column C, which is designed to assign the highest possible rank to tied values, the outcome for Doug and Eric is straightforward. Since 95 is the top score, and two students achieved it, both students were assigned a rank of **1**. This immediately

signals that they are co-leaders in the assessment.

The critical consequence of this ranking choice is the skipping of rank positions. Because ranks 1 and 2 were effectively consumed by the two tied scores, the next highest score, 90 (received by Fiona), is assigned the rank of **3**. This jump from 1 to 3 is characteristic of the standard **RANK** function and must be considered when calculating percentiles or distributing awards based on rank order. This method is the most traditional way of handling ties, prioritizing the designation of 'best' over sequential order.

Method 2: RANK.AVG (Average Rank)

The **RANK.AVG** function, applied in column D, provides a middle path by calculating the average rank positions consumed by the tied scores. Since Doug and Eric tied for the 1st and 2nd positions, the function calculated the average: $(1 + 2) / 2 = 1.5$. Consequently, both Doug and Eric were assigned a rank of **1.5**.

This fractional rank clearly indicates that neither student achieved the absolute rank of 1 alone, yet they are mathematically positioned between the first and second best possible outcomes. Following this, the next score (Fiona's 90) correctly assumes the rank of **3**. The advantage of **RANK.AVG** is its use in scenarios where maintaining the mathematical integrity of the rank distribution is paramount, such as in advanced statistical analysis where ranks are converted back into indices or used in comparison tests. It provides a non-subjective, statistically neutral solution to the tie.

Method 3: RANK + COUNTIF (Sequential Rank)

The composite formula using **RANK** and **COUNTIF** (in column E) is the definitive method for breaking ties and enforcing a unique, sequential rank for every single data point, even identical ones. This requires introducing a secondary criterion for ordering, which, in this case, is the row position within the dataset.

The formula first identifies the score of 95 as rank 1. When it processes Doug (the first person with 95, appearing in row 2), the **COUNTIF(B\$2:B2, B2)** component counts only one occurrence (itself), resulting in an adjustment of $1 - 1 = 0$. Doug keeps the rank of **1**. Since Doug appeared first in the dataset he received a rank of **1** and Eric then received a rank of **2**. This method assigned a rank of **1** to the first highest value that occurred and a **2** to the highest value that occurred next.

Conclusion: Choosing the Right Ranking Strategy

The choice between the three ranking methods in Google Sheets depends entirely on the analytical objective and the interpretation required for tied data. There is no universally 'correct' method; rather, there is the method most suited to your data narrative.

If your priority is to strictly recognize all maximum values equally and accept rank gaps, use the standard **RANK** function. If your objective requires a statistically fair, balanced rank that preserves the total rank sum and results in fractional ranks, **RANK.AVG** is appropriate. Finally, if you absolutely require a unique, gap-less sequential ranking based on the order of appearance, the composite formula leveraging **RANK** and COUNTIF is essential.

Mastering these three techniques empowers users to handle complex tie scenarios with precision, ensuring that the final output from your array analysis accurately reflects the intended criteria.

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

ARABPSYCHOLOGY.COM