

Learning to Rank Data by Multiple Columns in Google Sheets

Authored by
Mohammed loot

October 31, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Learning to Rank Data by Multiple Columns in Google Sheets*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=7250>

Introduction: Why Multi-Criteria Ranking is Essential

When analyzing complex datasets, standard ranking methods often fall short. Using the basic [RANK function](#) in [Google Sheets](#) typically considers only a single metric. However, in real-world scenarios--such as evaluating employee performance, ranking academic scores, or analyzing sports statistics--a single metric rarely provides a complete picture. Data analysts frequently encounter situations where two or more items share the exact same value in the primary ranking column. This requires a sophisticated solution to implement effective [tie-breaking](#) rules, ensuring that every data point receives a unique, precise rank.

To overcome the limitations of single-column ranking and accurately assign ranks when ties occur, we must employ a combined formula utilizing both the **RANK** function and the powerful [SUMPRODUCT function](#). This specific combination allows us to establish a hierarchy: ranking first by the most important column, and then using subsequent columns as tie-breakers in a specified order of precedence. This approach transforms a potentially ambiguous ranking into a definitive and reliable metric for data evaluation.

The following syntax provides a robust methodology for assigning ranks based on multiple criteria simultaneously. This particular construction is designed to rank items in a descending manner (highest value equals Rank 1) based first on the values found in Column B, and subsequently using Column C to resolve any resulting ties. Mastering this formula is crucial for anyone performing detailed statistical analysis within the spreadsheet environment.

```
=RANK(B2,$B$2:$B$11)+SUMPRODUCT(--($B$2:$B$11=$B2),--(C2<$C$2:$C$11))
```

Understanding the Core Google Sheets Ranking Formula

The solution above, while seemingly complex, is composed of two distinct and complementary parts. The initial component, `=RANK(B2, B2:B11)`, performs the standard ranking calculation based solely on the primary column (Column B). This part assigns the initial rank, but it will inevitably assign the same rank number to any rows that share identical values in B2. This is where the secondary component, the **SUMPRODUCT** function, becomes indispensable, acting as the intelligent tie-breaker mechanism.

The [SUMPRODUCT function](#) is typically used to multiply corresponding components in given arrays or ranges and returns the sum of those products. In this specific application, however, we leverage its capability to handle [Array formulas](#) without requiring the traditional Ctrl+Shift+Enter entry. The structure `--(B2:B11=$B2)` checks for rows where the value in Column B is equal to the current row's value, returning an array of TRUE/FALSE values. The double unary operator (`--`) then coerces these logical values into numerical values (1 for TRUE, 0 for FALSE).

The second array within the **SUMPRODUCT**, `--(C2<:C2:C11)`, is the actual tie-breaker. For all rows identified as ties in the primary column (where the first array returned 1), this section checks how many of those tied rows have a value in Column C that is *greater* than the current row's value in C2. Since we are aiming for a descending rank (higher values are better), every time a tied row has a higher secondary value, it signifies that the current row should be pushed down the ranking order by one spot. The result of **SUMPRODUCT** is thus an integer representing the exact number of rows that tied the primary score but exceeded the secondary score, which is then added to the base rank.

Practical Example: Ranking Basketball Players by Points and Assists

To illustrate the practical application of this powerful ranking technique, consider a dataset containing performance metrics for a group of basketball players. Our goal is to rank these players from best (Rank 1) to worst (Rank 10). The primary metric for determining the best player is **Points**. If two players achieve the same number of points, the tie must be broken by the secondary metric, **Assists**, where the player with more assists is considered superior.

Suppose we have the following dataset in Google Sheets, detailing the points and assists compiled by ten different athletes over a season. This scenario perfectly highlights the need for multi-criteria ranking, as ties in the Points column are highly likely and must be resolved definitively to assign a unique rank to each player.

	A	B	C	D
1	Player	Points	Assists	
2	A	25	8	
3	B	25	5	
4	C	20	4	
5	D	20	7	
6	E	18	9	
7	F	18	9	
8	G	16	4	
9	H	14	3	
10	I	11	4	
11	J	11	3	
12				
13				
14				
15				
16				
17				
18				

In this example, Column B contains the **Points** (the primary ranking criteria), and Column C contains the **Assists** (the secondary, tie-breaking criteria). Our objective is to generate a final ranking list in Column D, where Rank 1 is assigned to the player with the highest overall performance score, incorporating both metrics according to our defined priority.

Implementing the Descending Rank Formula

We will apply the combined **RANK** and **SUMPRODUCT** formula in cell D2 and then drag it down to apply to the rest of the players (cells D2:D11). The formula is designed to calculate the player's base rank using **Points**, and then incrementally adjust that rank based on how many players who tied on points had a better score in **Assists**.

We enter the following exact formula into cell D2:

```
=RANK(B2,$B$2:$B$11)+SUMPRODUCT(--($B$2:$B$11=$B2),--(C2<$C$2:$C$11))
```

Notice the use of absolute references (**\$B\$2:\$B\$11** and **\$C\$2:\$C\$11**) for the ranges. This is critical because when the formula is copied down, the reference range must remain fixed, while the references to the current row (**B2** and **C2**) must change dynamically to **B3**, **C3**, and so on. After

applying the formula across the entire dataset, the resulting ranks reveal the precise performance order, successfully resolving all initial ties.

The following screenshot demonstrates the successful execution of this formula, resulting in a unique ranking for every player, from Rank 1 (best) through Rank 10 (worst). This visual confirmation validates that the tie-breaking logic successfully incorporated the secondary criteria.

D2 *fx* =RANK(B2,\$B\$2:\$B\$11)+SUMPRODUCT(--(\$B\$2:\$B\$11=\$B2),--(C2<\$C\$2:\$C\$11))

	A	B	C	D	E	F
1	Player	Points	Assists	Ranking		
2	A	25	8	1		
3	B	25	5	2		
4	C	20	4	4		
5	D	20	7	3		
6	E	18	9	5		
7	F	18	9	5		
8	G	16	4	7		
9	H	14	3	8		
10	I	11	4	9		
11	J	11	3	10		
12						
13						
14						
15						
16						
17						
18						

Handling Tie-Breakers and Interpreting Results

The effectiveness of this ranking method is best understood by examining how it handles specific ties. The core function of the **SUMPRODUCT** segment is to accurately increment the base rank only when necessary, ensuring that superior tie-breaking scores are properly rewarded with a higher overall rank.

Consider the instance where two or more players tied for the highest number of points. For example, if we look at Players A and B in the provided data, they both scored the maximum number of points, resulting in an initial tie for Rank 1. The formula then executes the tie-breaking logic:

The base rank is calculated using the **Points** column.

The [SUMPRODUCT](#) component identifies all players tied on points.

For Player A, the formula checks how many of those tied players have *more* assists than Player A. If the answer is zero, the rank remains 1.

For Player B, the formula checks how many of those tied players have *more* assists than Player B. Since Player A had more assists, the count returns 1, meaning Player B's rank is incremented by one spot ($1 + 1 = 2$).

Therefore, because Player A had a higher value in the tie-breaking **Assists** column compared to Player B, Player A receives the definitive ranking of **1**, while Player B is assigned the ranking of **2**. This systematic approach ensures that the highest performer, based on the established hierarchy (Points then Assists), is always correctly positioned at the top of the list.

Adjusting for Ascending Order Ranking

In certain analytical contexts, it may be necessary to rank data in **ascending** order, meaning the lowest value receives Rank 1, and the highest value receives the highest rank. For instance, if we were ranking marathon times, the fastest time (lowest number) would be the best. Converting our multi-criteria formula to handle ascending order requires two specific modifications to the original structure.

First, we must modify the base [RANK function](#) to rank in ascending order by adding the optional argument **1**. Second, because the tie-breaker logic in **SUMPRODUCT** is still counting how many tied scores are higher than the current value, we must adjust the final result to correctly reflect the ascending order structure. This is accomplished by adding **+1** to the end of the entire formula, effectively shifting the zero-based tie-break count into the correct rank position.

If you wish to rank the players such that Rank 1 is the worst performer and Rank 10 is the best performer (ascending order based on points, then assists), you would use the following modified formula:

```
=RANK(B2,$B$2:$B$11, 1)+SUMPRODUCT(--($B$2:$B$11=$B2),--(C2<$C$2:$C$11))+1
```

The resulting ranks are inverted, with the worst performer now possessing a rank of **1** and the best performer receiving the maximum rank of **10**. This modification provides the flexibility needed to apply multi-criteria ranking regardless of whether the definition of "best" corresponds to the highest or the lowest numerical value in the dataset.

The following screenshot demonstrates the application of the ascending rank formula:

D2 fx =RANK(B2,\$B\$2:\$B\$11, 1)-SUMPRODUCT(--(\$B\$2:\$B\$11=\$B2),--(C2<\$C\$2:\$C\$11))+1

	A	B	C	D	E	F
1	Player	Points	Assists	Ranking		
2	A	25	8	10		
3	B	25	5	9		
4	C	20	4	7		
5	D	20	7	8		
6	E	18	9	6		
7	F	18	9	6		
8	G	16	4	5		
9	H	14	3	4		
10	I	11	4	2		
11	J	11	3	1		
12						
13						
14						
15						
16						
17						
18						
19						

Conclusion and Advanced Ranking Considerations

Implementing multi-criteria ranking in Google Sheets using the combined **RANK** and **SUMPRODUCT** method is an essential skill for advanced data manipulation. This technique moves beyond simple sorting by providing a definitive, unique numerical position for every entry, even when multiple values are identical in the primary sorting column. This rigorous approach ensures data integrity and clarity in analytical reports.

While we demonstrated a two-criteria ranking (Columns B and C), this methodology can be extended to include three or more tie-breaking columns by simply adding additional arrays to the **SUMPRODUCT** function. For instance, to include a third tie-breaker column (Column D), you would append a third array to the **SUMPRODUCT** component, ensuring that the hierarchy is strictly maintained (e.g., `--(D2:D11=$D2)`). It is critical to ensure that all criteria are properly tested and weighted according to their importance in the overall ranking structure.

Mastering this formula provides a powerful tool for complex data analysis, allowing analysts to derive meaningful insights from datasets that require nuanced evaluation. We encourage further exploration of related functions to enhance your spreadsheet proficiency.

Additional Resources

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