Techni "Kal" ki Tayari...

MASTERING THE UNUSUAL: A DEEP DIVE INTO A LITTLE-KNOWN EXCEL FORMULA'S APPLICATIONS

Devansh Kariya
Email : devanshj7number@gmail.com

Introduction

In the realm of professional skills, individuals with proficiency in Microsoft Excel are universally valued. As businesses increasingly rely on data-driven decision-making, the ability to navigate and contrive spreadsheets has become an essential asset. It's not just about knowing the basics; it's about exploring the nuances of Excel's capabilities, especially those concealed in the tabs seldom visited. Among these, our focus today is on an uncommon Excel formula that can change the approach towards application of formulas – 'AGGREGATE'. Let's explore its application through a practical example, demonstrating how mastering such formulas can streamline tasks and contribute to overall efficiency.

Illustration

Imagine you are tasked with organizing an auction for a cricket tournament. Your goal w.r.t. use of excel is to create a workbook wherein details of auction such as number of teams, wallet balance, players list with base price, etc are recorded prior to auction and a dynamic spreadsheet wherein list of players bought by each team and wallet balance keeps updating automatically during auction.

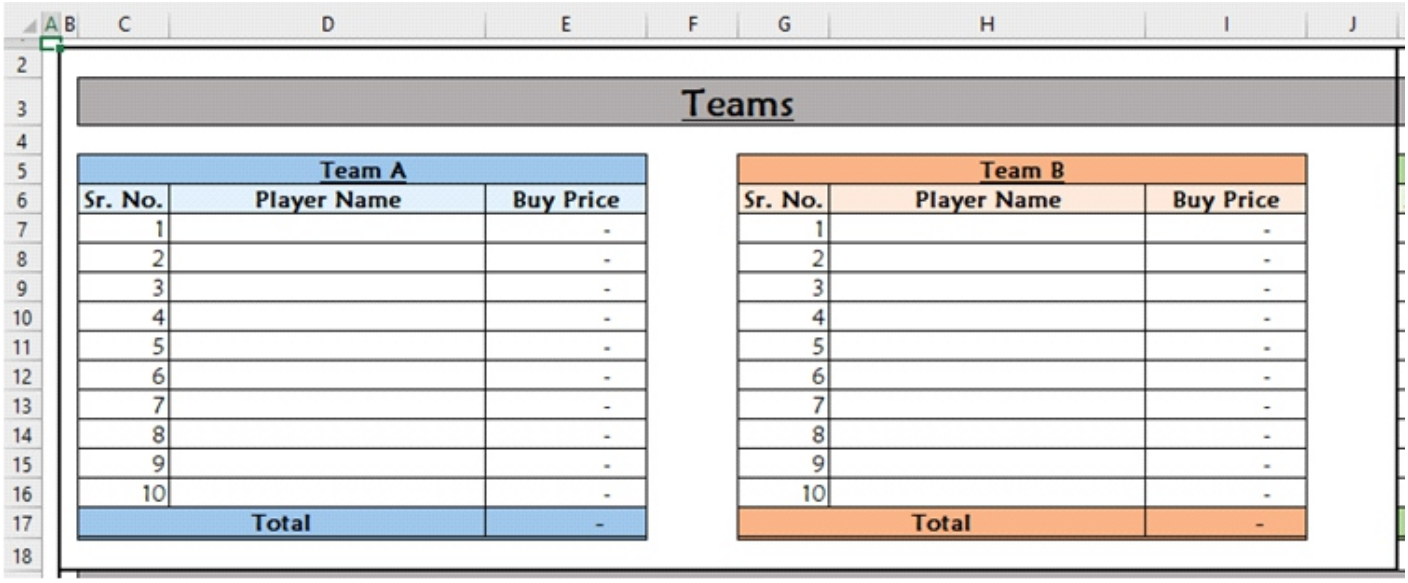
Solution

Details of wallet limit, players list, base prices, etc. may be entered manually in simple tables format. Then the main task – creating a dynamic spreadsheet, for this let's consider creating two spreadsheets as follows;

1. Auction Entry – for recording auction transactions.

Round	Player No.	Player Name	Base Price	Team	Buy Price
1			-		
2			-		
3			-		
4			-		
5			-		

2. Teams – dynamic tables dependent upon 'Auction Entry' sheet.



Teams

Team A		
Sr. No.	Player Name	Buy Price
1		-
2		-
3		-
4		-
5		-
6		-
7		-
8		-
9		-
10		-
Total		-

Team B		
Sr. No.	Player Name	Buy Price
1		-
2		-
3		-
4		-
5		-
6		-
7		-
8		-
9		-
10		-
Total		-

In the 'Auction Entry' sheet – Player No. shall be entered manually during auction and corresponding player name and base price can be obtained by use of simple VLOOKUP. Once the player is sold, Team that bought it and buying price of player needs to be recorded manually. Drop down list (using data validation) can be used for ease in recording Team name.

Now, we need 'Teams' Sheet to get updated automatically based on the above-mentioned auction entry sheet. Here we will need 2 columns/details:

1. Buy Price – It is the price at which player is bought by the team owner. It can be obtained by simply using VLOOKUP wherein Lookup value is the Player Name.

2. Player Name – It can be obtained by using the FILTER (array, include, [if_empty]) formula. In this, 'array' refers to the column (even more than 1 is possible) from where output is required (player name column). 'Include' refers to the condition (team name = "Team A"). The complete formula will look like -

   =FILTER('Auction Entry'!D:D,'Auction Entry'!F:F=$C$5,"").

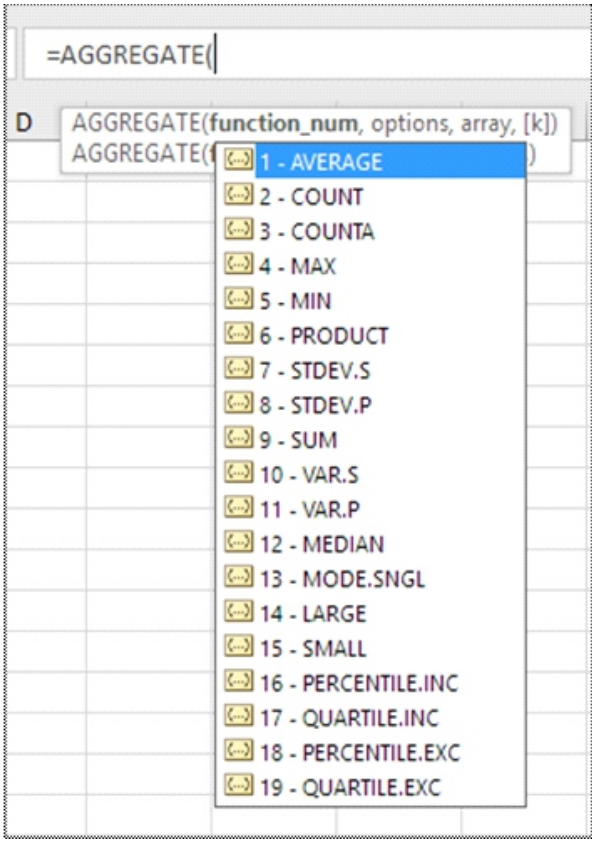
However, this 'FILTER' formula is available only on paid Microsoft 365. This limits the access to this amazing formula.

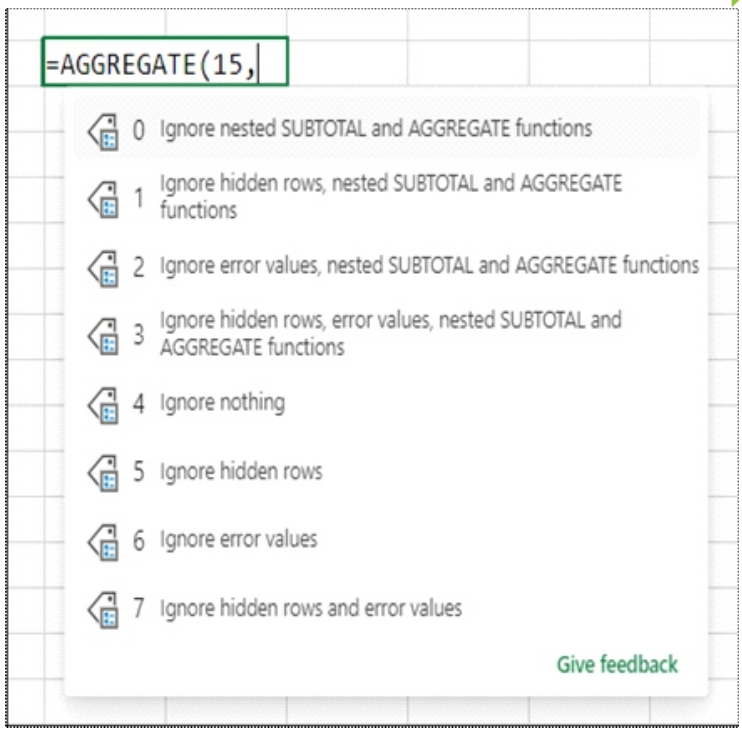
In order to overcome this, we can use 'INDEX' formula nested with 'AGGREGATE' and 'ROW' formulas as follows;

1. INDEX (array, row_num, [column_num]) – Returns a value or reference of the cell at the intersection of a particular row and column, in a given range.

   In this formula, we can give reference of column 'D:D' as 'array' to get the player's name. As there is only 1 column in array, no need to enter any column_num in syntax. But we need to provide the 'row_num' from which it is supposed to pick-up player's name.

2. AGGREGATE (function_num, options, array, [k]) – This hidden gem has the capability of performing various functions along with customization options. Combinations of these functions and options can not only substitute but also overcome the limitations of formulas such as SUM, SUMPRODUCT, SMALL, LARGE, SUBTOTAL, FILTER, COUNT, COUNTA, MIN, MAX, AVERAGE, etc.





In current case, we would use function number 15 i.e. Small (returns the smallest value from a dataset) along with option number 7 i.e. Ignore hidden rows and error values from the dataset.

Now, Let's give our formula it's array using Row formula – Row formula returns the row number of the cell given as reference for example =ROW(C38) will the give the output as '38'. Similarly, when an entire array is given as reference it provides the row number for each cell for example =ROW(B32:B39) will give an array of output as '32, 33, 34, 35, 36, 37, 38, 39'.

We all know that, in MS excel, TRUE outputs are valued as '1' and FALSE outputs are valued as '0'.

Based on the above, we can create an array of row numbers of only those rows which satisfy our given condition and for rows that do not satisfy the condition - error values.

For example, =ROW('Auction Entry'!$F:$F)/('Auction Entry'!$F:$F=Teams!$C$5)

This will give the output as (1,2,3,4,…)/(1,0,0,1,…) i.e., all Row numbers divided by '1' if corresponding Team name in column F is Team A and 0 otherwise.

The final outcome for this formula will be an array as follows;

'1, #DIV/0! , #DIV/0! , 4, #DIV/0! , 6, 7, #DIV/0! , #DIV/0! , #DIV/0! , 11,…'

Now, as we had used option 7 in AGGREGATE formula – error values will get ignored giving an array as '1, 4, 6, 7, 11,…' and then function 15 i.e., Small will select the smallest value i.e., 1 in this case. Consequently. INDEX formula will pick-up value present in 1st row of our given array i.e. Player's name. In this way, we will get our desired output.

However, in the above-mentioned formula only the smallest value will be picked up. To overcome this, we need to enter 'K' value (function_num, options, array, [k]) i.e., degree/constant. For example, if K is 3 – AGGREGATE formula will provide the 3rd smallest row number. In order to make 'K' dynamic we can give reference to Sr. No.

Upon nesting the entire formula with IFERROR the complete formula will be as follow;

```
=IFERROR(INDEX('Auction Entry'!$D:$D,AGGREGATE(15,7,ROW('Auction Entry'!$F:$F)/('Auction Entry'!$F:$F=Teams!$C$5),$C7)),"")
```

Now, this single formula can be used for any number of teams just by changing reference of Team Name i.e., cell C5 in our case and we will get our dynamic spreadsheet.

Based on the above, we can get auto updating wallet balance and available surplus using SUMIFS and COUNTIFS formulas.

You can scan the QR code alongside to get a view of the auction format as per above-mentioned solution.

Alternative formats can be created by application of different formulas and creativity.

Please note: This array functions may not be directly recognized by MS Excel versions 2016 or lower. In order to use such array formulas, after typing the formula in the cell press 'Control + Shift + Enter' instead of 'Enter'. This adds {} at the start and end of formulas so that excel recognizes the array. For example {=ROW(C5:C12)}.

Conclusion

As we conclude, it's essential to underscore the broader importance of continually honing our spreadsheet skills. In the fast-paced professional landscape, adaptability and efficiency are paramount. Advanced Excel proficiency not only simplifies daily tasks but also positions individuals as valuable assets in their respective roles. The discovery and application of unusual formulas are emblematic of a proactive approach to skill development. By staying curious and embracing the less-explored functionalities of Excel, we not only enhance our individual capabilities but contribute to a more data-savvy and productive professional environment. So, let's keep refining our Excel skills, exploring new formulas, and unleashing the full potential of this indispensable tool in our professional toolkit.
