Learning to Filter Data in Google Sheets with Custom Formulas

Authored by

Mohammed looti

October 29, 2025

In the vast world of data management and analysis, Google Sheets stands as an exceptionally powerful and universally accessible tool, favored by professionals and casual users alike for its collaborative capabilities and intuitive interface. While standard filtering mechanisms--such as filtering by value, color, or simple text containment--offer foundational data organization, they often prove insufficient when dealing with complex, multi-criteria requirements. This is precisely the point where the advanced functionality of custom formulas becomes indispensable, unlocking highly specific and dynamic data manipulation capabilities that transform raw data into actionable insights.

The ability to integrate a custom formula directly into the filtering process, using the built-in **Filter by condition** function, represents a significant leap in spreadsheet mastery. This feature allows users to move beyond simple selection criteria, enabling them to define intricate logical rules based on calculations, comparisons across multiple columns, or sophisticated pattern matching. By utilizing this functionality, you significantly extend your ability to manage, cleanse, and analyze information within your spreadsheets, efficiently isolating precisely the data segments required for specialized reporting or deeper investigation, criteria that conventional filters simply cannot accommodate.

This comprehensive guide is designed to demystify the process of custom filtering. We will thoroughly walk through a practical, real-world example, meticulously demonstrating how to leverage this versatile function to refine complex datasets within Google Sheets. By the end of this tutorial, you will possess the knowledge required to construct and implement your own complex filtering logic, dramatically increasing your efficiency and analytical precision.

Understanding Custom Formulas for Filtering

At its core, a **custom formula** in Google Sheets filtering system functions as a logical evaluator applied sequentially to every single row in your selected data range. Instead of relying on predefined interface options--like "Show only rows where the text starts with 'A'"--you construct a formula using standard spreadsheet functions that must ultimately resolve to either the boolean value `TRUE` or `FALSE` for each corresponding row. If the calculation for a specific row yields `TRUE`, that row satisfies the condition and remains visible; conversely, if the calculation returns `FALSE`, the row is filtered out and hidden from view. This fundamental mechanism provides unparalleled flexibility for complex data requirements.

The true value of utilizing custom formulas becomes apparent in scenarios where simple column-by-column comparisons are inadequate. Imagine needing to identify transactions that occurred on a weekend AND exceeded a certain value, or finding employees whose tenure is greater than five years BUT whose job title does not contain "Manager." These intricate, compound criteria necessitate the logical precision afforded by combining multiple functions (like `AND`, `OR`, `IF`)

within a single filtering rule. Custom formulas provide the essential control required to effectively tackle such advanced filtering challenges with high efficiency and accuracy.

Furthermore, one of the most sophisticated and powerful tools frequently integrated into custom filtering is the REGEXMATCH function. This function facilitates advanced text processing by allowing for regular expression pattern matching, moving far beyond simple string searching. This capability is crucial when you need to match data that adheres to complex, variable structures--such as specific phone number formats, varying email addresses, or specific code sequences--rather than relying solely on exact or partial literal strings. When deployed in conjunction with powerful logical functions, such as the OR function or the `AND` function, REGEXMATCH enables the creation of filtering conditions that are truly sophisticated and highly dynamic.

Practical Example: Filtering Player Data

To fully grasp the practical application of custom filters, let us examine a specific, common scenario often encountered in data analysis. Consider a spreadsheet where you are tasked with managing a comprehensive dataset containing essential information about various basketball players. This dataset is structured into columns, typically including the players' names and their performance scores over a specified period. The challenge lies in quickly and reliably identifying only those players whose names carry a specific qualifier, such as "Good" or "Great," indicating their performance rating or classification.

Standard filtering options would demand a cumbersome, multi-step process for this objective. You would first have to filter for "text contains 'Good'," then clear that filter, and then filter for "text contains 'Great'," or perhaps rely on creating temporary helper columns. This method is inefficient, prone to error, and fails to provide a single, consistent view of the combined results. The goal here is distinct: we need a mechanism that checks for multiple, unrelated partial text matches simultaneously across the same column, presenting only the combined results in a single, filtered view.

The inherent limitations of basic filtering make this scenario an exemplary case for implementing a custom formula. By using a single logical expression, we can instruct Google Sheets to evaluate if a player's name meets either condition--containing "Good" OR containing "Great"--in one efficient operation. This capability ensures precise data isolation, dramatically reducing manual effort and processing time, particularly when working with spreadsheets containing thousands of records.

Suppose our initial dataset looks like this, prior to the application of any filtering logic:

	A	B	C	D
1	**Player**	**Points**		
2	Good Player	20		
3	Great Player	30		
4	Average Player	14		
5	Bad Player	7		
6	Great Player	26		
7	Good Player	22		
8	Average Player	16		
9	Average Player	12		
10	Bad Player	4		
11	Good Player	19		
12	Average Player	13		
13				
14				
15				
16				
17				
18				
19				
20				

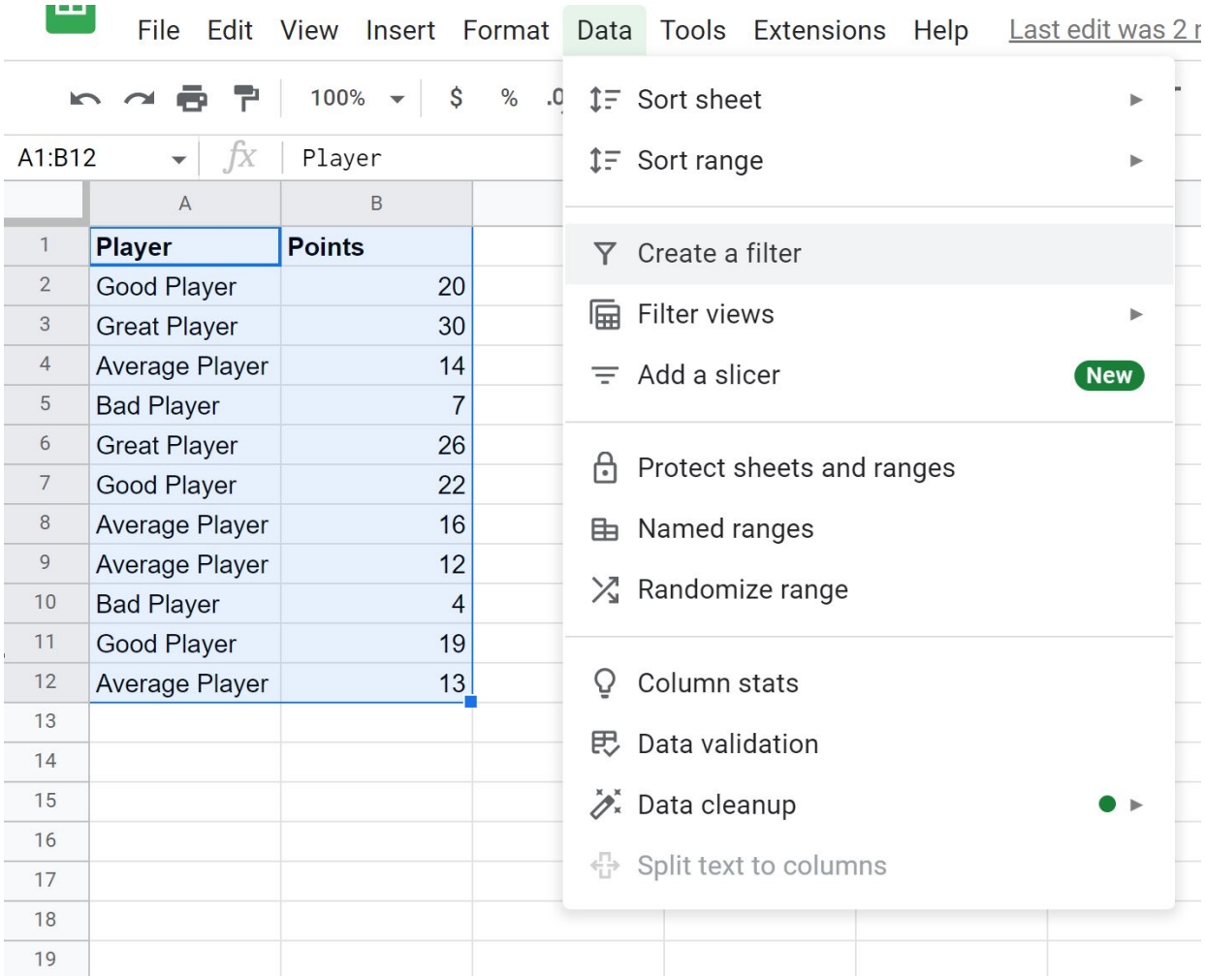
Step-by-Step Guide to Applying a Custom Filter

Implementing a custom filter in Google Sheets is a sequential process that ensures your complex logic is correctly applied to the desired scope of data. Follow these meticulously detailed instructions to successfully integrate your custom filtering logic and isolate the target rows in your spreadsheet.

Select Your Data Range: The foundational step involves accurately defining the scope of data to which the filter will be applied. Start by highlighting the entirety of the cell range that encompasses your complete dataset, including the header row. In the context of our player data example, this crucial range is designated as **A1:B12**. Ensuring the correct range is selected from the outset is absolutely paramount, as the filter functionality will only evaluate data contained within these specified boundaries, and neglecting the header row might cause it to be filtered erroneously.

Access the Filter Functionality: Once the data range is highlighted, navigate your cursor to the **Data** tab, which is prominently located along the top ribbon menu of the Google Sheets interface. Clicking this tab will reveal a dropdown menu containing various data management options. From this menu, select the command **Create a filter**. Executing this command will instantly place distinct

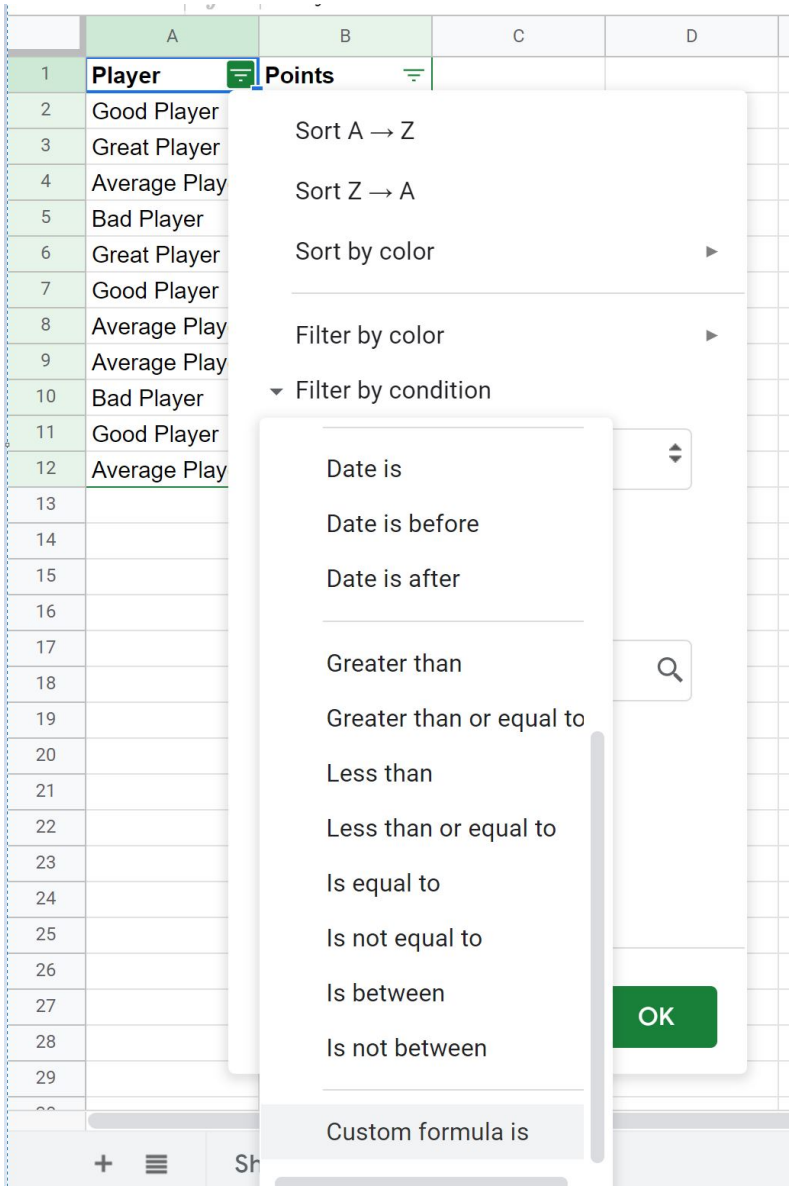
filter icons--typically represented by a small inverted triangle or funnel symbol--onto the right side of every column header within your previously selected range, indicating the filter is active across the dataset.



Choose 'Filter by condition': Proceed by clicking the newly visible filter icon associated with the specific column you intend to apply the logical condition to. In our scenario, since we are analyzing player names, you must click the filter icon located next to the **Player** column header (Column A). A comprehensive filter menu will pop up, offering sorting options and various filtering criteria. Within this menu structure, carefully locate and hover your mouse over the section titled **Filter by condition**, which manages rule-based filtering rather than simple value selection.

Select 'Custom formula is': From the extended list of preset conditions that appear under the 'Filter by condition' section, scroll all the way down to the bottom of the list. Here, you will find the most flexible option labeled **Custom formula is**. Click on this option. This action is crucial, as it transforms the filter panel, presenting a dedicated, open text box where you are required to input your specific, self-defined filtering logic. This text box is where the precise, boolean-evaluating

custom formula will reside.



Crafting the Custom Formula for Text Matching

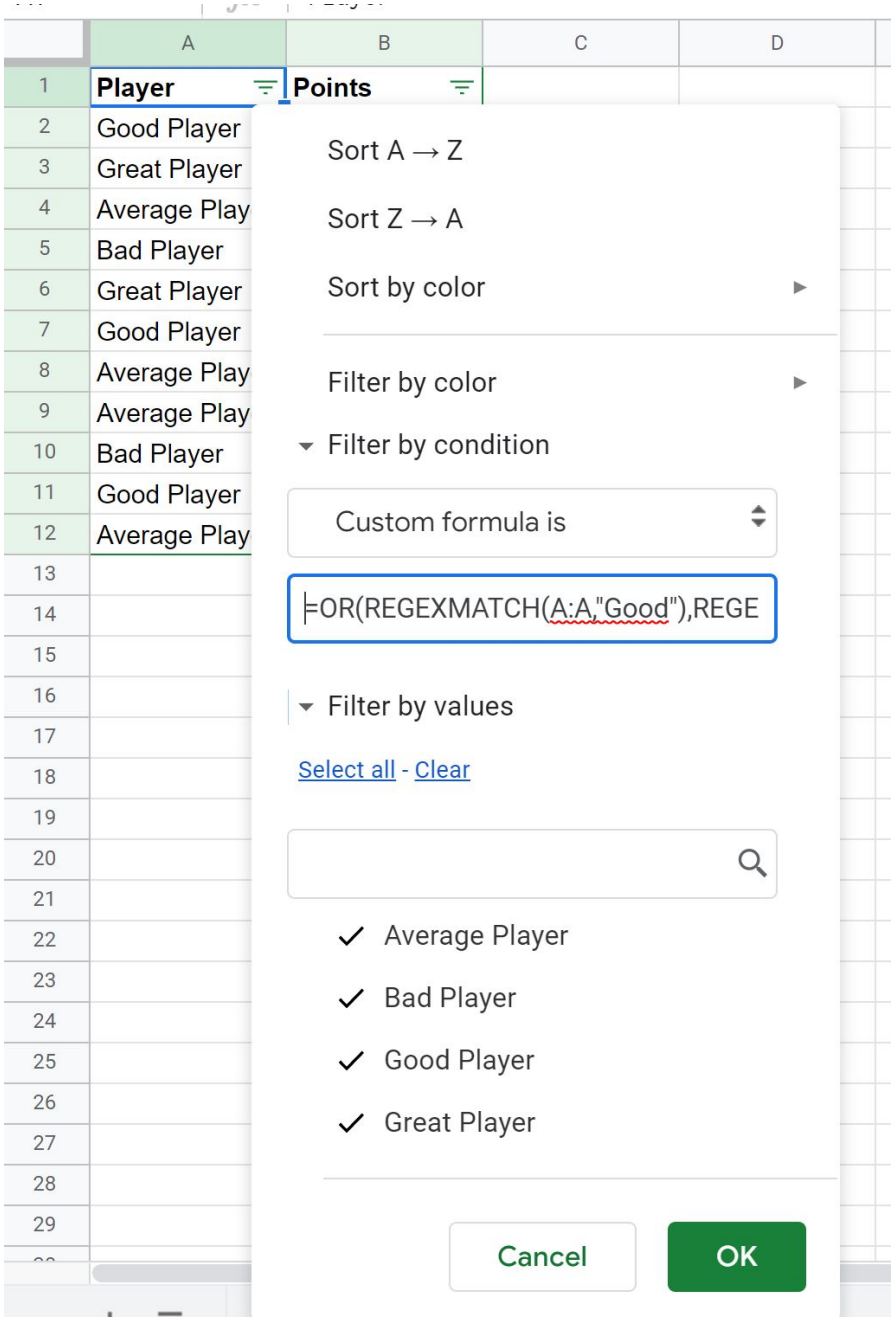
Having successfully navigated to the "Custom formula is" input field, the next critical phase involves accurately constructing the logical expression that meets our complex criteria. Our specific objective is to display rows where the Player column contains the string "Good" or the string "Great." To achieve this necessary 'either/or' logic combined with partial text searching, we must employ a powerful combination of the OR function and the robust REGEXMATCH function. The formula must be written referencing the first cell of the column we are filtering, ensuring it applies correctly across all subsequent rows.

Input the following formula precisely into the designated custom formula text box:

=OR(REGEXMATCH(A:A,"Good"),REGEXMATCH(A:A,"Great"))

Understanding the architecture of this formula is vital for troubleshooting and future customization. The formula operates through nested logic, where the internal functions provide criteria checks, and the outer function determines the final boolean result for the filtering engine. The process ensures that for every row evaluated, the system checks if at least one of the internal conditions holds true. This sophisticated structure is the key to executing complex, multi-criteria filtering effectively.

OR(...): This is the primary logical wrapper function. The OR function evaluates all the logical expressions contained within its parentheses. Crucially, it returns `TRUE` immediately if *any* of the individual conditions return `TRUE`. This is essential for our goal, as we want to include a player if they are rated "Good" OR if they are rated "Great." If neither condition is met, the `OR` function returns `FALSE`, and the row is hidden.
REGEXMATCH(A:A, "Good"): The REGEXMATCH function is deployed here to handle the partial text search. It performs a check to see if any part of the text string within the specified column matches the provided regular expression pattern. In this simplified case, the pattern is the literal string "Good". Because REGEXMATCH inherently looks for matches anywhere within the text, it successfully identifies rows like "The Good Player" without needing additional wildcard characters, returning `TRUE` if a match is detected.
A:A Column Reference: When a full column reference like `A:A` is used within a filter's custom formula, Google Sheets automatically adjusts the reference for each row it evaluates. For example, when checking row 5, the system effectively evaluates `REGEXMATCH(A5, "Good")`. This automatic relative referencing across the column is fundamental to making the filter formula work correctly across the entire dataset.

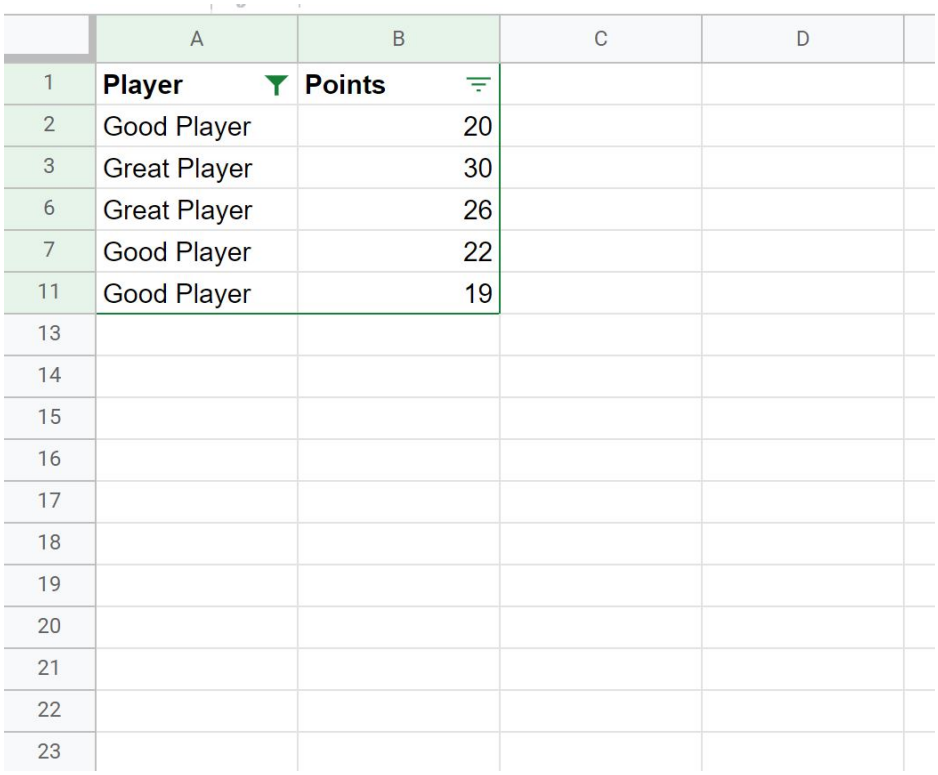


Observing the Filtered Results

Upon entering the complete formula and clicking the **OK** or **Apply** button within the filter panel, the transformation of your data should be instantaneous. Google Sheets immediately processes the custom formula against every row in the defined range, applying the logical condition and visually adjusting the spreadsheet to display only the rows that satisfy the criteria. Rows that returned `TRUE` (those containing either "Good" or "Great") remain visible, while all others are concealed, providing immediate, precise visual feedback.

This result powerfully illustrates the efficiency and precision afforded by using a custom formula. Instead of resorting to complex manual sorting, repetitive application of multiple standard filters, or the laborious use of temporary columns, a single, concise, and well-structured formula instantly achieves the desired, complex outcome. This method is particularly invaluable when managing large-scale datasets, where manual filtering would be not only impractical but also introduce a high risk of overlooking crucial data points.

The successful application of the filter confirms that the power of logical functions like REGEXMATCH, combined using the OR function, allows for robust text-based querying that replicates advanced database search capabilities directly within your spreadsheet environment. You now have a clean, focused subset of your data ready for further analysis or reporting, streamlined by custom logic.



	A	B	C	D
1	**Player**	**Points**		
2	Good Player	20		
3	Great Player	30		
6	Great Player	26		
7	Good Player	22		
11	Good Player	19		
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				

Expanding Your Filtering Capabilities

The example utilizing the OR and REGEXMATCH functions serves as a foundational demonstration, yet the true scope of custom formula filtering in Google Sheets is exponentially greater. By integrating knowledge of other standard spreadsheet functions, you can construct filtration rules for virtually any conceivable data scenario, moving beyond simple text comparisons to incorporate time-based logic, mathematical calculations, and inter-column relationships. This mastery is the gateway to utilizing Sheets as a truly dynamic data analysis platform.

Developing proficiency in writing effective custom formulas empowers you to perform highly specific and remarkably efficient data analysis, transforming daunting organizational tasks into routine, automated procedures. Consider the potential for managing complex inventories, tracking project timelines, or segmenting customer data based on multiple dynamic factors. By deeply understanding the underlying TRUE/FALSE evaluation logic and actively experimenting with various function combinations, you can unlock advanced filtering capabilities that will radically enhance your overall productivity and analytical depth within the spreadsheet environment.

The flexibility of custom formulas allows you to apply logic that would otherwise require scripting or external tools. Examples of advanced uses include:

Filtering by **dynamic date ranges** using time-sensitive functions such as `TODAY()`, `WEEKDAY()`, or `EDATE()`. For example, `=A:A > TODAY() - 7` filters for entries made in the last week.
Applying **numerical conditions based on calculations** involving multiple columns, such as `=AND(B:B>10, B:B<20)` to precisely find scores that fall within a defined range between 10 and 20 exclusively.
Identifying **blank or non-blank cells** accurately using built-in validation functions like `ISBLANK(C:C)` or `NOT(ISBLANK(C:C))`, which is vital for data cleaning and integrity checks.
Implementing **comparisons across different columns** in the same row, such as filtering for rows where the value in Column B is greater than the value in Column C, using a simple formula like `=B:B > C:C`.

Additional Resources

To further expand your proficiency in Google Sheets and continue your journey toward mastering advanced data handling techniques, we encourage you to explore these related tutorials that delve into other essential and complex data operations: