

How to Easily Perform What-If Analysis in Google Sheets

Authored by
stats writer

December 3, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Perform What-If Analysis in Google Sheets*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=103884>

What-if analysis in Google Sheets is a fundamental modeling technique that empowers users to systematically explore various operational scenarios and predict their corresponding outcomes. This sophisticated form of sensitivity testing moves beyond simple data observation, allowing business analysts and decision-makers to manipulate key input variables and instantly observe the resulting impact on dependent metrics. It is an indispensable tool for strategic planning, budgeting, and forecasting, providing a clear path to understanding the relationship between inputs and outputs in complex data models.

The basic framework of this analysis involves defining a set of assumptions (the "inputs") and using established formulas to calculate a desired result (the "output"). While simple manual modification of cells can achieve basic testing, true what-if analysis leverages powerful built-in functions to automate scenario exploration. Key functions utilized often include conditional logic like the **IF** function, aggregated conditional calculations such as **SUMIFS** and **COUNTIFS**, and lookup mechanisms like **VLOOKUP**. By dynamically linking these functions to changeable input cells, users can build robust models that react instantaneously to potential changes in market conditions, costs, or sales volumes.

Furthermore, enhancing the user interface of the analysis through tools like data validation significantly improves usability. Utilizing data validation to create controlled dropdown menus and input rules ensures that the data driving the scenarios remains clean and relevant. This prevents errors and allows stakeholders to easily switch between predefined scenarios, such as "Best Case," "Worst Case," and "Most Likely," thereby controlling the parameters of the analysis and ensuring reliable simulation outcomes.

Core Functions for Scenario Modeling

While the ultimate goal of many what-if analyses is finding a specific input value (which requires advanced tools like Goal Seek), the foundation relies heavily on core spreadsheet functions. Understanding how to structure these functions is critical for defining the relationship between inputs and outputs. For instance, the **IF** function allows the model to switch behaviors based on a threshold--e.g., calculating different tax rates if revenue exceeds a certain level. This conditional branching is essential for modeling realistic business rules and complexities that define the scenario space.

More advanced scenarios often require summarizing data based on multiple criteria. This is where conditional aggregation functions like **SUMIFS** and **COUNTIFS** become invaluable. Imagine needing to calculate total revenue only from products sold in a specific region during a certain quarter; these functions handle such complexity efficiently. By linking the criteria cells (Region, Quarter) to input cells, analysts can instantly change parameters and view the resulting aggregated figures, providing granular control over the scenario testing process.

Ultimately, the power of a well-designed what-if model lies in its flexibility. By systematically structuring the inputs, calculations, and output metrics, users create a living dashboard that allows for seamless exploration of countless possibilities. However, sometimes the question is inverted: instead of asking "What happens if I change X?", we ask "What input (X) is required to achieve a target outcome (Y)?" This reverse engineering of the data model is where standard functions fall short and dedicated tools are needed.

The Necessity of Goal Seeking for Reverse Analysis

As previously defined, what-if analysis traditionally involves adjusting inputs to observe outputs. However, business problems often require the inverse approach, known as Goal Seek or "back-solving." This technique is necessary when a specific target outcome--such as reaching a target profit margin or achieving a certain level of total revenue--is fixed, and the analyst needs to determine the exact value of a single input variable required to meet that target. Manually iterating through potential input values until the target is hit is time-consuming, prone to error, and rarely achieves the exact required solution.

Consider a scenario where a marketing team needs to achieve \$100,000 in sales next month, and they know their conversion rate. Manually guessing the required advertising spend to hit that target is inefficient. Goal Seek automates this iterative process by using algorithms to rapidly converge on the precise input value. It identifies how a single changing cell must be altered to force a formula cell to equal a specific numerical goal. This capability transforms the spreadsheet from a prediction tool into a prescriptive strategic planner.

In Google Sheets, unlike some desktop spreadsheet applications, this powerful function is not built into the core menu structure. Instead, users must rely on a specialized third-party add-on to unlock this advanced computational capability. The following step-by-step example shows how to perform this exact goal-seeking type of what-if analysis.

Example Scenario Setup: Calculating Target Revenue

To illustrate the application of Goal Seek, we will use a common business modeling example involving sales and revenue calculation. Suppose a retail operation sells three distinct products (A, B, and C), each with a fixed unit price. The current total revenue is calculated based on the number of units sold for each product.

The structure of our data model is straightforward: we multiply the Units Sold by the Price per Unit for each product to get the Product Revenue, and then sum these individual revenues to determine the Total Revenue. The following image represents our current dataset in Google Sheets:

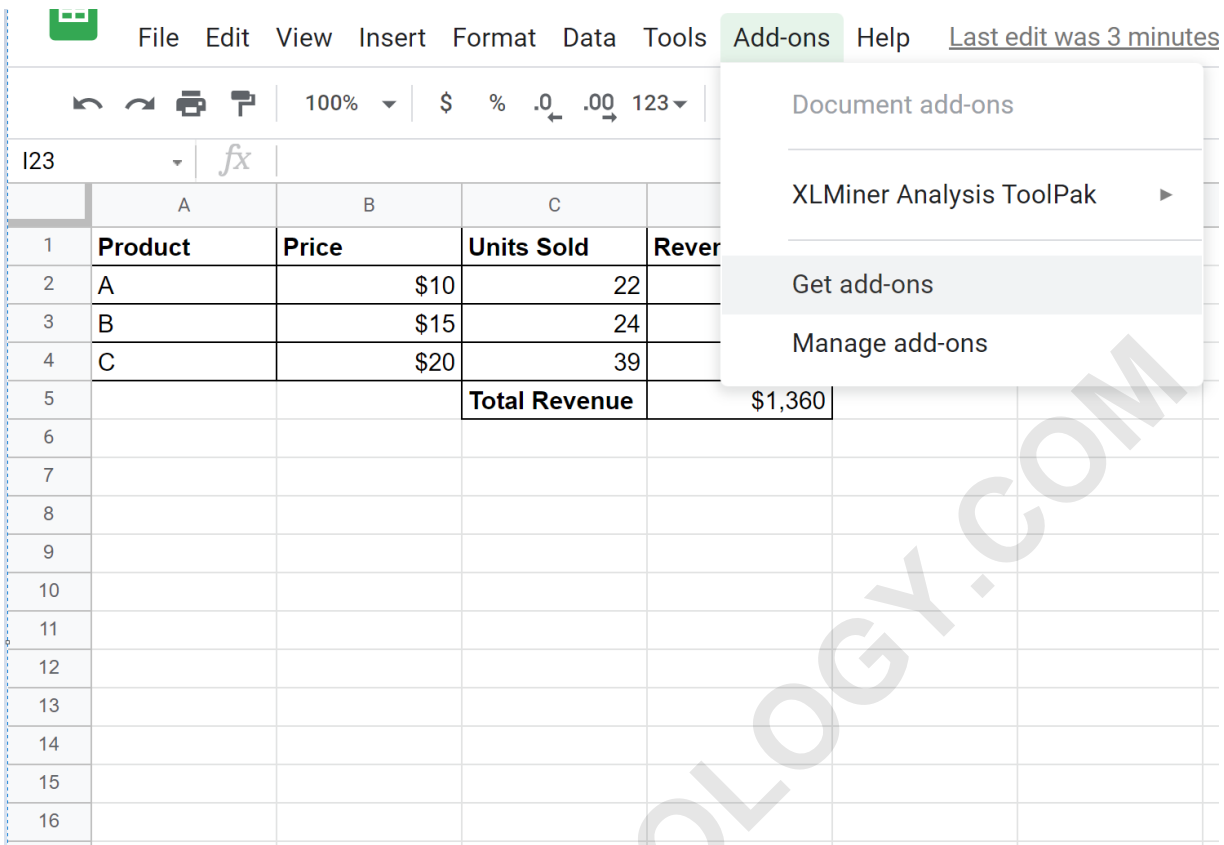
	A	B	C	D	E
1	Product	Price	Units Sold	Revenue	
2	A	\$10	22	\$220	
3	B	\$15	24	\$360	
4	C	\$20	39	\$780	
5			Total Revenue	\$1,360	
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					

The central question in this scenario is a prescriptive one: The store manager has set a specific target of reaching a total revenue figure of **\$2,000** for the period. We need to determine precisely how many additional units of **Product A** must be sold to meet this ambitious financial goal, assuming the sales volumes of Product B and Product C remain constant. This requires finding the input value in cell D2 that forces the calculated Total Revenue in cell D5 to equal \$2,000.

Step 1: Installing the Goal Seek Add-On

Since the Goal Seek functionality is not native to the standard Google Sheets environment, the initial step requires obtaining the necessary extension from the Google Workspace Marketplace. This add-on, typically titled **Goal Seek for Sheets**, provides the algorithms required for performing efficient reverse calculation. It is essential to complete this installation process before attempting any advanced reverse what-if analysis.

To initiate the installation, navigate to the main menu bar within your spreadsheet and click the **Add-ons** tab. From the resulting dropdown menu, select the **Get add-ons** option. This action will launch the dedicated marketplace interface, allowing you to search for and manage external tools that integrate with your spreadsheet environment.



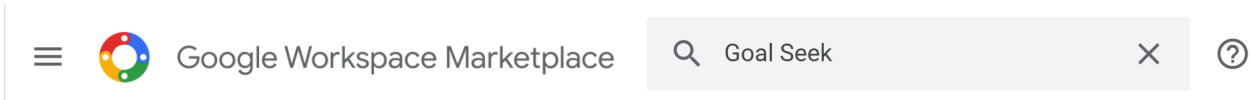
The screenshot shows the Google Sheets interface with the 'Add-ons' menu open. The spreadsheet data is as follows:

	A	B	C	
1	Product	Price	Units Sold	Rever
2	A	\$10	22	
3	B	\$15	24	
4	C	\$20	39	
5			Total Revenue	\$1,360
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				


The 'Add-ons' menu is open, showing the following options:


- Document add-ons
- XLMiner Analysis ToolPak
- Get add-ons
- Manage add-ons

Once the marketplace window is open, utilize the search bar to locate the tool by typing "Goal Seek." Identify and select the appropriate listing, usually labeled **Goal Seek for Sheets**, which is the necessary component for this process. After selecting the correct result, proceed to click the **Install** button. Be prepared to review and accept the permissions requested by the application, as these are necessary for the add-on to interact with your sheet data and formulas. Upon successful acceptance, the **Goal Seek** tool will be integrated into your **Add-ons** menu, ready for immediate use.



Search results for Goal Seek

Goal Seek for Sheets 

Published by  Google Cloud

Goal Seek
google.com
The Goal Seek add-on for Google Sheets enables users to calculate the required value of...

★ 2.6 • ↓ 1,545,445

Ponder

What-if Analysis Tools

Ponder
[Sheetsify](https://sheetsify.com)
What-if analysis tools (goal seek & data tables) for Sheets.

★ 3.8 • ↓ 948

Step 2: Configuring and Executing the Goal Seek Analysis

With the add-on successfully installed, the execution of the reverse what-if analysis begins by launching the dedicated tool. Return to the **Add-ons** menu, locate **Goal Seek** in the list of installed extensions, and select **Open**. This action will typically display a sidebar panel on the right side of your Google Sheets interface, providing the required input fields for defining the analysis parameters.

Format Data Tools **Add-ons** Help [Last edit was 8 minutes ago](#)

\$ % .0 .00 123 ▾

	c	
	Units Sold	Rever
0	22	
5	24	
0	39	
	Total Revenue	

Document add-ons

Goal Seek ▶

XLMiner Analysis ToolPak ▶

Get add-ons

Manage add-ons

Open

Help

The Goal Seek panel requires three critical pieces of information to define the calculation:

Set cell: This is the cell containing the formula whose result we want to control. In our example, this is cell **D5** (Total Revenue).

To value: This is the specific target number we wish the formula cell to achieve. We input **2000**, representing the desired total revenue of \$2,000.

By changing cell: This is the single input cell that the Goal Seek algorithm will iteratively adjust until the target value is met. We input cell **D2** (Units Sold of Product A).

Carefully input these parameters into the panel fields, ensuring cell references are accurate. Once configured, initiate the computational process by clicking the **Solve** button.

Goal Seek

Settings

Set Cell *
D5

Cell of the Target value

To Value *
2000

Target value

By Changing Cell *
D2

Cell to be changed when finding Target value

Options

Solve



Analyzing the Results and Iterations

Upon clicking **Solve**, the Goal Seek algorithm executes a rapid series of calculations. It systematically iterates by testing various values in the specified changing cell (D2) until the set cell (D5) converges precisely on the target value (\$2,000). This process eliminates the guesswork and manual trial-and-error traditionally associated with reverse engineering financial models.

The result of this sophisticated computation reveals the necessary input value. In our specific retail example, the output shows that to achieve a total revenue of \$2,000, the store must sell approximately **860 units of Product A**. The calculated value in cell D2 may be a decimal (e.g., 859.3 units), which should be interpreted realistically based on the context of the input variable--in this case, rounding up to 860 units would be required to exceed or meet the \$2,000 target. The final state of the spreadsheet reflecting this solution is shown below, confirming the precise correlation between the required input and the targeted output:

	A	B	C	D	E
1	Product	Price	Units Sold	Revenue	
2	A	\$10	22	859.9999619	
3	B	\$15	24	\$360	
4	C	\$20	39	\$780	
5			Total Revenue	1999.999962	
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					

A significant benefit of utilizing the Goal Seek add-on is the detailed reporting on the solution process. The analysis panel provides metrics regarding the computational effort expended to find the solution. For instance, the result summary indicates the total time elapsed (e.g., 20.3 seconds) and the number of **iterations** (e.g., 48) the algorithm performed. This information is valuable for complex models, allowing users to gauge the efficiency of the calculation and understand the computational complexity involved.

 **Solve Status** 

Start Time: 11/1/2021, 9:27:22 AM
Status: **Solution found**
Sheet Name: Sheet1
Set Cell: D5
To Value: 2000
By Changing Cell: D2
Iterations: 48
Running Time: 20.3 seconds
Solution: 859.9999618530273

It is important to note the configurable limits within the Goal Seek tool. By accessing the **Options** button, users can customize constraints, such as setting the maximum number of seconds or iterations to use until Goal Seek stops running. By default, these limits are often set to **200 iterations** and **120 seconds**, respectively. Adjusting these parameters can be crucial when dealing with extremely large datasets or highly complex formulas, ensuring the process terminates gracefully even if an exact solution is difficult to find. This robust configuration capability ensures that the what-if analysis remains efficient and resource-conscious.