Data Analysis with Excel

This next lesson builds on the previous lesson where we cleaned and formatted data. Now we prepare the data for analysis. The primary tool for this in Excel is the Pivot Table.

This lesson shows how to create a pivot table to summarize or organize data in Excel.

Often, when looking at data, you want to summarize or reorganize the data. The PivotTable tool in Excel is a popular way to reorganize or group data by certain fields.

A pivot table is a general term for any table that can be quickly "pivoted" to different summary views of data. The spelling PivotTable is specific to Excel's pivot table tool name.

Pivot table

A type of table that can be used to calculate, summarize, and analyze data belonging to multiple categories.

Summary table

A concise visual summary of one or more aggregations and calculations performed on a larger dataset.

Summary tables

A pivot table is a fancy type of summary table. But what is a summary table? A summary table aggregates your data (or groups the items) to show an organized snapshot of the data.

To get a sense of what summary tables do, imagine that you have a catering business. The first spreadsheet below shows the orders that you need to deliver to your clients. The second spreadsheet shows this data organized into a summary table showing how many orders you need to deliver each day. Summary tables highlight valuable information or groupings that you may not otherwise notice in the data.

	A	B	C
1	**Client**	**Delivery Date**	**Quantity**
2	A	8/20/2021	3
3	B	8/20/2021	4
4	C	8/22/2021	5
5	D	8/20/2021	9
6	E	8/20/2021	4
7	F	8/22/2021	6
8	G	8/24/2021	8
9	H	8/24/2021	4

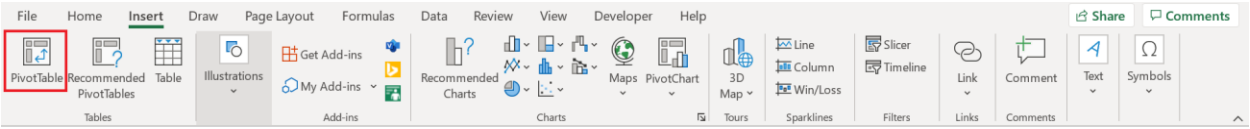
Delivery Date	Quantity
8/20/2021	20
8/21/2021	0
8/22/2021	11
8/23/2021	0
8/24/2021	12
Total	**43**

Pivot tables

A pivot table is an enhanced version of a summary table. Pivot tables are used heavily in data analysis because you can quickly update or rearrange this type of summary table in just a few seconds!

Creating a PivotTable

Excel's PivotTable tool can be found in the Insert tab of the ribbon.



On the right side of the pivot table sheet, there is a PivotTable Fields pane. Here, you can drag fields of your data to Filters, Columns, Rows, or Values. Where you drag fields determines how your report looks.

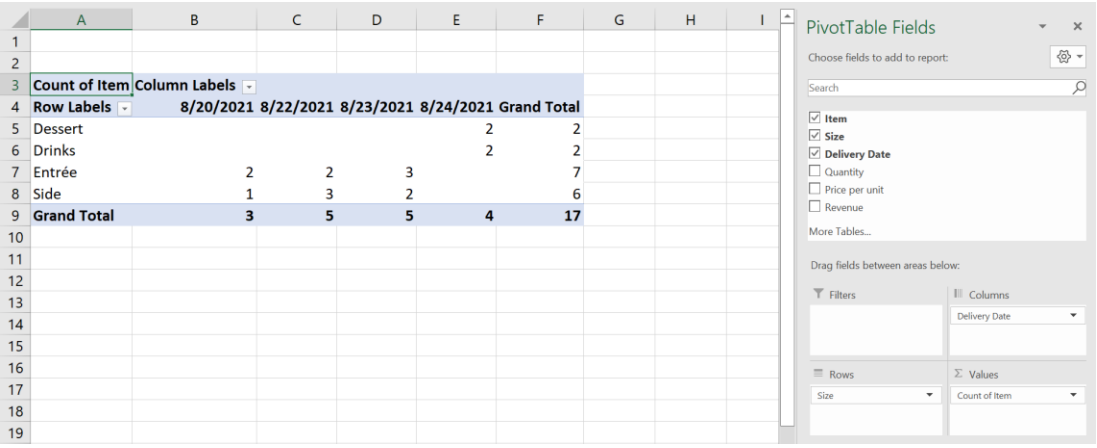
Use Case 1:

Say you want to reorganize the catering data to show how many items are required for each date and for each Size. Because you want to see the data grouped by Size, drag the Size field into the Rows section. This uses all possible values of Size to create the rows of your pivot table.

Next, you also want to see how many items are required per date, so drag the Delivery Date field into the Columns section. This uses all possible values of Delivery Date to create columns for your pivot table.

Finally, the values that you want to see are the number of items per date per size, so drag the Item field to the Values section. Excel is smart and automatically calculates the count of the Item field per date per size.

The resulting pivot table is shown below.



Count of Item	Column Labels				
Row Labels	8/20/2021	8/22/2021	8/23/2021	8/24/2021	Grand Total
Dessert				2	2
Drinks				2	2
Entrée	2	2	3		7
Side	1	3	2		6
Grand Total	3	5	5	4	17

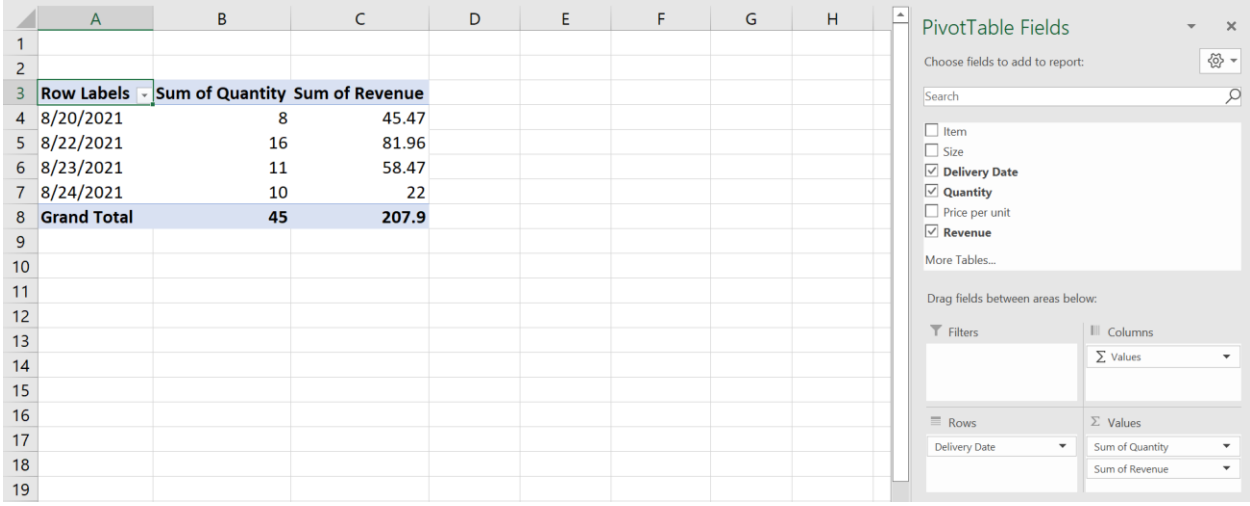
Use Case 2:

What if you wanted to see total quantities and total revenues per delivery date, so that you know exactly how much to prepare for each date and how much you can expect to earn off those deliveries? Because you want the quantities and revenue per Delivery Date, this means that you want the data grouped by Delivery Date. So, drag the field Delivery Date to the Rows section. The pivot table uses all possible values of Delivery Date to create the rows for your pivot table.

Next, you want to see the total quantities for each date. Because a total is a value, drag the Quantity field to the Values section. This organizes the quantities next to each Delivery Date. Excel is smart; it knows that Quantity is a numeric field and that you likely want a total or sum. So, it automatically produces the sum of Quantity next to each date.

Finally, you want to see the total revenue for each date. Because a total is a value, drag the Revenue field to the Values section. This automatically produces the sum of Revenue next to each date.

This pivot table is shown below.



Row Labels	Sum of Quantity	Sum of Revenue
8/20/2021	8	45.47
8/22/2021	16	81.96
8/23/2021	11	58.47
8/24/2021	10	22
Grand Total	**45**	**207.9**

In the examples above, Excel sensed that the fields Quantity and Revenue are numeric data types, so dragging them into the Values section means that you probably wanted the totals or sums of each. Similarly, when you dragged the Item field into the Values section, it sensed that because Item is a string data type, you probably wanted the total count or number of items. You can also customize the Values section further. If you click the drop-down for the field in the Values section and then click Value Field Settings, you will see that there are many options for calculations that you can perform on that field.

Maybe you just want to see the largest order of an item for each date. Then you would select the Max option from the list. Or maybe you just want to see the average revenue generated for each date, rather than the total. Then you would select the Average option from the list. Both of these selections are reflected in the pivot table below.

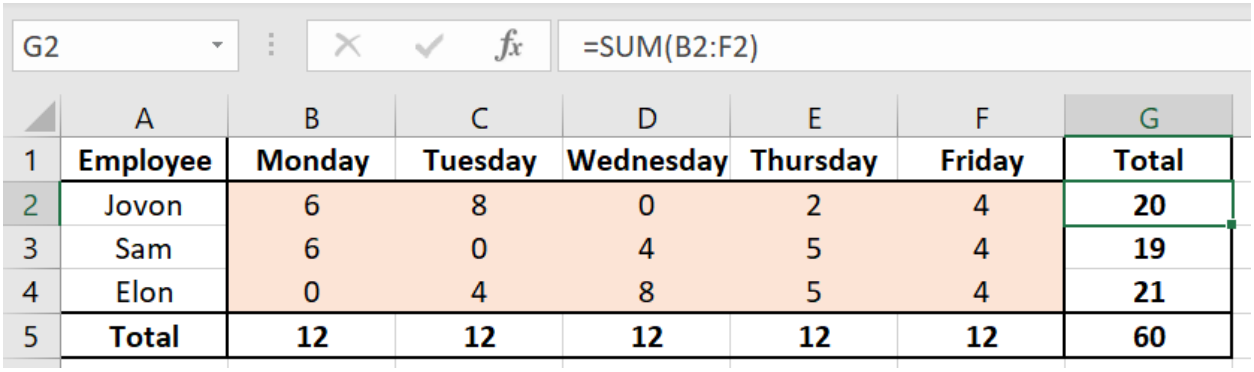
Dynamic formulas

This lesson explains how to create dynamic formulas in Excel using everything you've learned so far in this course. Dynamic formulas allow you to change inputs to test different scenarios so that you don't have to create a new formula each time.

So far in this program, you've created many formulas using Excel functions. Some have been simple formulas, such as SUM, and some have been complex formulas, such as nested INDEX and MATCH functions. In this lesson, you'll create dynamic formulas using functions that you've already learned.

A dynamic formula is one that uses cell references in a way that lets a user change the value of inputs and immediately see the new output. This is an efficient and concise way to test multiple values in a formula without having to rewrite the formula over and over again for each new input.

It's often helpful to use many dynamic formulas together. Say you're a manager at a company, and you have three people directly reporting to you. You're trying to plan out their schedule hours for the upcoming week, calculating how many hours each person works and how many total hours are worked each day. The example below shows the schedule for the upcoming week. The total row along the bottom and the total column at the end of the table are dynamic formulas that calculate totals using the SUM function and cell references. For example, the total in cell G2 is a sum of cells B2 through F2 (which you can write as B2:F2), so B2:F2 are inputs to the dynamic formula. If any of these values are changed, the formula recalculates based on the new values.



	A	B	C	D	E	F	G
1	**Employee**	**Monday**	**Tuesday**	**Wednesday**	**Thursday**	**Friday**	**Total**
2	Jovon	6	8	0	2	4	**20**
3	Sam	6	0	4	5	4	**19**
4	Elon	0	4	8	5	4	**21**
5	**Total**	**12**	**12**	**12**	**12**	**12**	**60**

Multistep dynamic formulas

The power of dynamic formulas can be seen with multistep dynamic formulas, which are dynamic formulas that perform multiple steps using multiple dynamic inputs.

In the scheduling example above, you also want to evaluate how many hours of overtime, if any, each employee accrues in the upcoming week. You can create a dynamic formula that checks if the value in column G is greater than 40 hours. If it is, the formula then calculates how many hours over 40 the employee is scheduled to work. This is shown below.

H2 =IF(G2>40, G2-40, 0)

	A	B	C	D	E	F	G	H
1	**Employee**	**Monday**	**Tuesday**	**Wednesday**	**Thursday**	**Friday**	**Total**	**Overtime Hours**
2	Jovon	8	8	8	7	10	**41**	**1**
3	Sam	8	9	7	6	9	**39**	**0**
4	Elon	8	9	7	8	10	**42**	**2**
5	**Total**	**24**	**26**	**22**	**21**	**29**	**122**	**3**
6								

This is a multistep dynamic model because the IF statement in H2 uses G2 as an input, and G2 uses B2:F2 as inputs. Adjusting any of the inputs in B2:F2 causes the total hours in G2 to adjust, which then causes overtime hours in H2 to adjust.

Say Jovon changes his hours so that he works only 6 hours on Wednesday and 5 hours on Thursday. These changes cause the automatic recalculation of his G2 total hours and his H2 overtime hours, as shown.

H2 =IF(G2>40, G2-40, 0)

	A	B	C	D	E	F	G	H
1	**Employee**	**Monday**	**Tuesday**	**Wednesday**	**Thursday**	**Friday**	**Total**	**Overtime Hours**
2	Jovon	8	8	6	5	10	**37**	**0**
3	Sam	8	9	7	6	9	**39**	**0**
4	Elon	8	9	7	8	10	**42**	**2**
5	**Total**	**24**	**26**	**20**	**19**	**29**	**118**	**2**
6								

Luckily, because you have this multistep dynamic formula set up, you don't have to rewrite the formula for total hours or for overtime hours when someone adjusts their schedule!

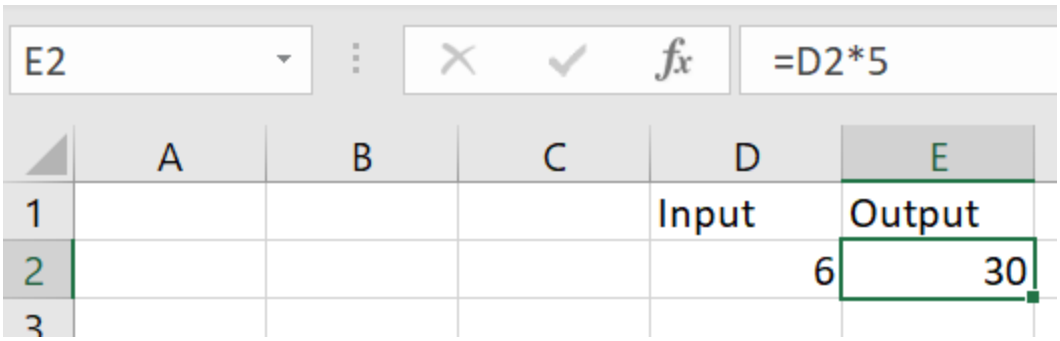
Goal-seeking

In the previous lesson, we learned how to create dynamic formulas to perform multiple calculations at one time. As you now know, dynamic formulas allow you to change inputs to see the resulting output immediately. But what if there's a specific output that you need, and you want to know what input value achieves this output? This is called goal-seeking or back-solving, and it's the process of starting out with a known output and trying to calculate the input needed to get that output.

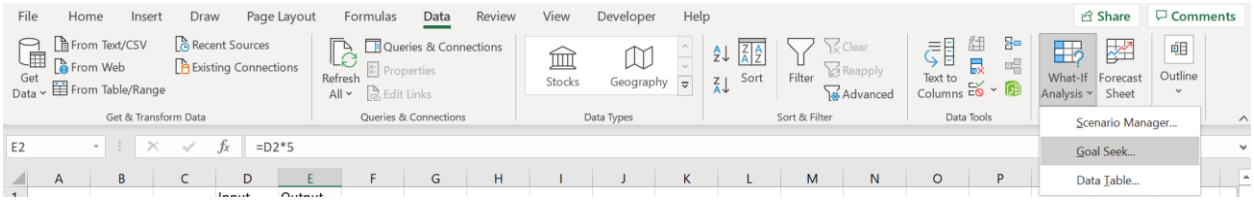
For example, if you have a certain budget set for the month, you can use back-solving to figure out how much you can spend on an upcoming trip while still meeting your monthly budget. Excel has a special tool that's designed to do this for you.

Using Goal Seek to back-solve calculations

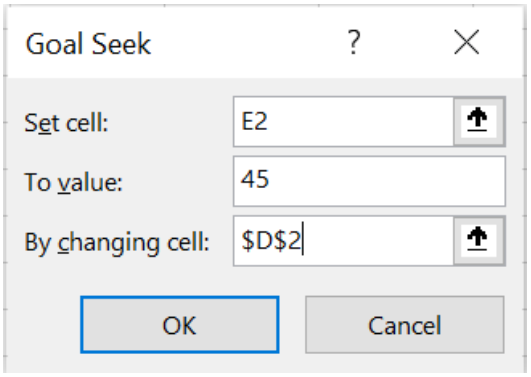
To start with a simple example, check out the formula below. It evaluates 6 (in cell D2) multiplied by 5 to get the output of 30.



Now, say you want to know what input value is needed to achieve an output of 45 instead of 30, using the same formula structure. Excel has a tool called Goal Seek. You can access this tool via the What-If Analysis button in the Data tab of the ribbon.

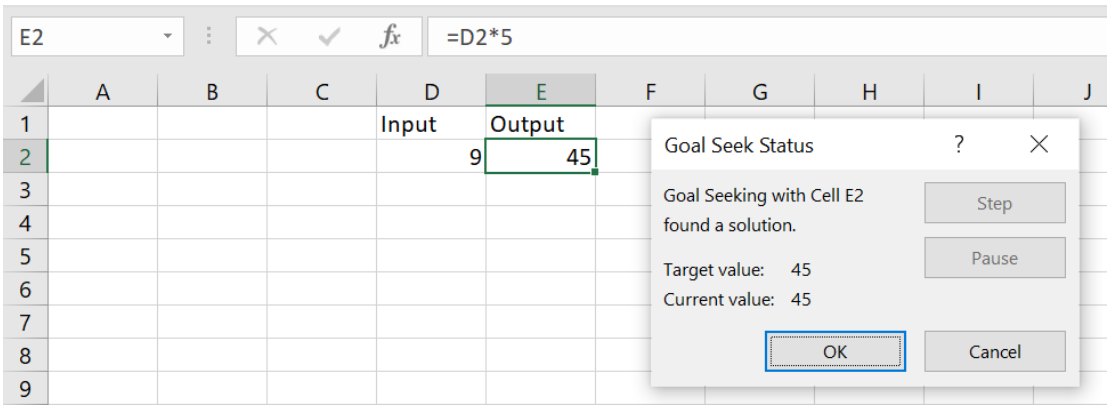


Click Goal Seek, and you will see the following dialog.



In the Set cell box, select or enter the cell that contains the output of the formula (E2, in this case). In the To value box, enter your desired output. In this case, the desired output is 45. In the By changing cell box, enter or select the input cell that you want Excel to solve for. The By changing cell box requires a cell that contains a single value, not a formula. In this case, you want Excel to find the D2 value that achieves the output of 45. Click OK.

You will then see one more dialog, letting you know the result of this goal seek: You set the target output value to 45, and Excel successfully found an input that achieves that. Cell E2 now shows the output that you set, 45. And cell D2 now shows the new input that was solved for, which is 9.

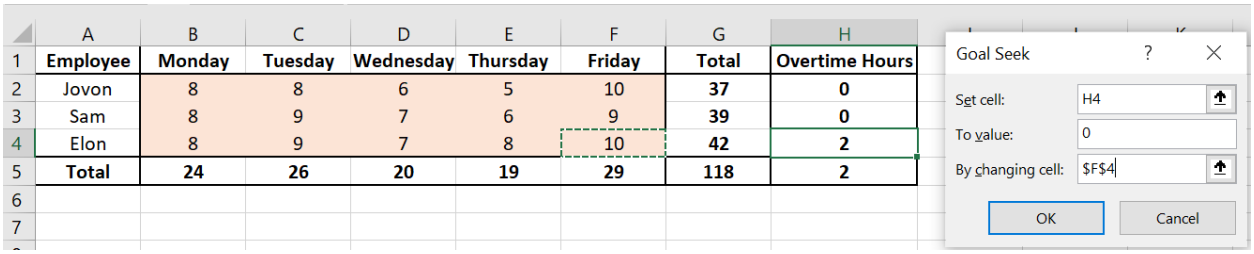


Next, revisit the example from the previous lesson to practice goal-seeking with a more complicated dynamic formula. Recall that this table shows the schedule of hours for each of your employees for the upcoming work week.

Employee	Monday	Tuesday	Wednesday	Thursday	Friday	Total	Overtime Hours
Jovon	8	8	6	5	10	**37**	**0**
Sam	8	9	7	6	9	**39**	**0**
Elon	8	9	7	8	10	**42**	**2**
Total	**24**	**26**	**20**	**19**	**29**	**118**	**2**

Elon is the only person currently scheduled for overtime, but he has requested to not work any overtime next week. He has flexibility on Fridays, so you want to adjust his Friday hours so that he doesn't work any overtime. You can use the Goal Seek tool to solve for the number of hours he must work on Friday to avoid any overtime.

To do this, click the What-If Analysis button and then Goal Seek.



Use H4 as the Set cell value, because that's the output that you're changing. Enter 0 in the To value box because you want zero overtime for Elon. Finally, set the By changing cell box to F4, or Elon's Friday hours. Click OK.