

# 3 Excel Functions That Will Make You a Spreadsheet Expert

SEQUENCE, LET, and LAMBDA are not commonly used, but they solve specific problems that would otherwise require complex workarounds or lengthy formulas that are difficult to consider.

Microsoft Excel has thousands of functions, but most people focus on the basics, such as SUM and AVERAGE. While these functions work well for simple tasks, there are three functions that handle complex situations with less effort. SEQUENCE, LET, and LAMBDA aren't as commonly used, but they solve specific problems that would otherwise require complex workarounds or lengthy formulas that are difficult to review.

## 4. The SEQUENCE function automatically generates data

### Generate dynamic number and date series

	A	B	C	D	E	F	G
1	Date	Region	Product Category	Salesperson	Reference	Units Sold	Sales Amount
2	1/5/2024	North	Electronics	John Smith	1001	15	3750
3	1/5/2024	South	Clothing	Sarah Johnson	1002	23	2300
4	1/5/2024	East	Home & Garden	Mike Wilson	1003	12	1800
5	1/5/2024	West	Sports	Lisa Brown	1004	18	2160
6	1/12/2024	North	Clothing	John Smith	1005	31	3100
7	1/12/2024	South	Electronics	Sarah Johnson	1006	8	2400
8	1/12/2024	East	Sports	Mike Wilson	1007	25	3000
9	1/12/2024	West	Home & Garden	Lisa Brown	1008	14	2100
10	2/3/2024	North	Sports	John Smith	1009	22	2640
11	2/3/2024	South	Home & Garden	Sarah Johnson	1010	19	2850
12	2/3/2024	East	Electronics	Mike Wilson	1011	11	3300
13	2/3/2024	West	Clothing	Lisa Brown	1012	28	2800
14	2/17/2024	North	Home & Garden	David Chen	1013	16	2400
15	2/17/2024	South	Sports	Emma Davis	1014	33	3960

SEQUENCE creates sequential arrays of numbers without having to manually enter each value. If you need a list of employee IDs, invoice numbers, or date ranges, this function makes it easy.

The syntax is very simple:

=SEQUENCE(rows, [columns], [start], [step])

?????Let's analyze the parameters:

1. **rows** : Specify how many numbers you want vertically.
2. **columns** : Controls horizontal distribution - leave blank if there is only one column.
3. **start** : Set the starting number, default is 1.
4. **step** : Specifies the distance between numbers, also defaults to 1.

For sales data sets, SEQUENCE is useful for generating reference numbers. For example, the following formula generates numbers from 1 to 32.

=SEQUENCE(32)

Similarly, if you need to start from 1001, you can use:

=SEQUENCE(32, 1, 1001)

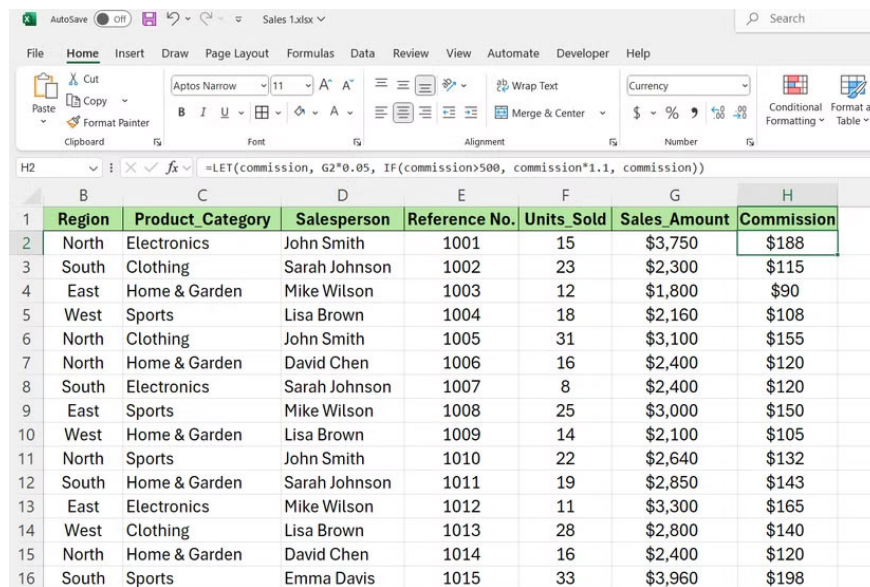
This formula is useful for date series. The following formula will generate 12 consecutive dates starting from January 1. This is better than manually entering dates for monthly reports or project milestones.

=SEQUENCE(12, 1, DATE(2025, 1, 1), 1)

You can also create just workdays by combining SEQUENCE with other DATE functions in Excel, such as WORKDAY, for more complex scheduling situations.

### 3. The LET function makes complex formulas easier to maintain

#### Eliminate repetitive calculations and improve readability



	Region	Product Category	Salesperson	Reference No.	Units Sold	Sales Amount	Commission
1	North	Electronics	John Smith	1001	15	\$3,750	\$188
2	South	Clothing	Sarah Johnson	1002	23	\$2,300	\$115
3	East	Home & Garden	Mike Wilson	1003	12	\$1,800	\$90
4	West	Sports	Lisa Brown	1004	18	\$2,160	\$108
5	North	Clothing	John Smith	1005	31	\$3,100	\$155
6	North	Home & Garden	David Chen	1006	16	\$2,400	\$120
7	South	Electronics	Sarah Johnson	1007	8	\$2,400	\$120
8	East	Sports	Mike Wilson	1008	25	\$3,000	\$150
9	West	Home & Garden	Lisa Brown	1009	14	\$2,100	\$105
10	North	Sports	John Smith	1010	22	\$2,640	\$132
11	South	Home & Garden	Sarah Johnson	1011	19	\$2,850	\$143
12	East	Electronics	Mike Wilson	1012	11	\$3,300	\$165
13	West	Clothing	Lisa Brown	1013	28	\$2,800	\$140
14	North	Home & Garden	David Chen	1014	16	\$2,400	\$120
15	South	Sports	Emma Davis	1015	33	\$3,960	\$198

The LET function assigns names to values in a formula. This eliminates repetitive calculations and makes the work easier to read. Instead of writing the same expression over and over again, you define it once and reference it by name.

The syntax follows this pattern:

```
=LET(name1, value1, [name2, value2, .], calculation)
```

You can define multiple variables by adding name-value pairs. The final calculation will use these named variables to produce the result.

Looking at a sales dataset, suppose you are calculating a salesperson's commission plus bonuses. Without LET, you would write:

```
=IF(G2*0.05>500, G2*0.05*1.1, G2*0.05)
```

The commission calculation  $B2*0.05$  appears twice. With LET, it becomes clearer:

```
=LET(commission, G2*0.05, IF(commission>500, commission*1.1, commission))
```

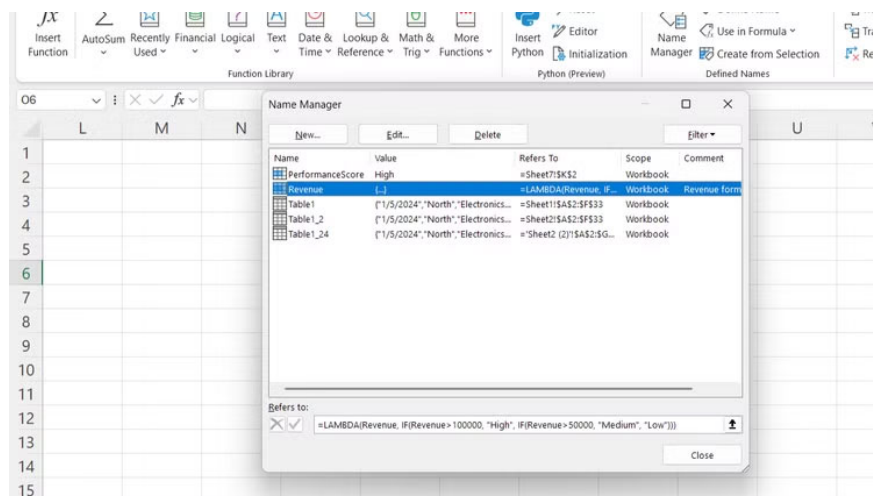
It does the same calculation but defines the "commission" once at the beginning. You only need to change the commission rate in one place.

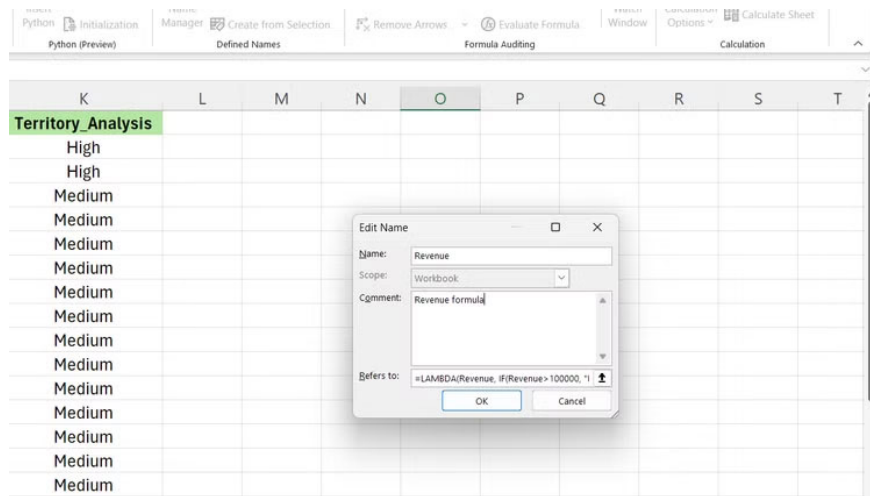
For complex profit margin analyses, LET is even more useful. The following example clearly defines each component.

```
=LET(revenue, G2, costs, L2, margin, (revenue-costs)/revenue, IF(margin>0.3, "High", "Low"))
```

## 2. LAMBDA functions create reusable custom functions

### Build custom functions for repetitive business logic





	G	H	I	J	K	L	M
1	<b>Sales_Amount</b>	<b>Commission</b>	<b>Profit_Margin</b>	<b>Revenue</b>	<b>Territory_Analysis</b>		
2	3750	187.5	High	116247	=Revenue(J2)		
3	2300	115	High	117093	High		
4	1800	90	High	65494	Medium		
5	2160	108	High	96541	Medium		
6	3100	155	High	92108	Medium		
7	2400	120	High	56123	Medium		
8	3000	150	High	85903	Medium		
9	2100	105	High	57358	Medium		
10	2640	132	High	61281	Medium		
11	2850	142.5	High	76190	Medium		
12	2200	110	High	54054	Medium		

LAMBDA creates custom functions that you can use multiple times in your workbook. Instead of copying formulas all over the place, you create a single function that accepts input and returns a calculated result.

The syntax is:

`=LAMBDA(parameter1, [parameter2, .], calculation)`

Parameters act as placeholders - when you call the function, you pass actual values in place of these placeholders. The calculation uses these parameters to produce output.

Suppose you regularly calculate weighted performance scores. You can create a LAMBDA function like this:

`=LAMBDA(sales, quota, weight, (sales/quota)*weight)`

It creates a reusable function that takes 3 inputs: Actual sales, sales quota, and a weighting factor. The function returns a weighted performance score by dividing sales by quota and multiplying by the weighting factor. Name this function "PerformanceScore" using Excel's Name Manager.

**Tip:** To name your LAMBDA function, go to **Formulas > Name Manager > New**.

You can now call this function anywhere in the workbook.

=PerformanceScore(B2, C2, 0.7)

This function calculates the performance score using the provided sales, quota, and weighting factor.

To analyze regions, you can build a function that classifies regions based on revenue:

=LAMBDA(revenue, IF(revenue>100000, "High", IF(revenue>50000, "Medium", "Low")))

This function classifies revenue into three levels: **High** for revenue above \$100,000, Medium for revenue between \$50,000 and \$100,000, and Low for revenue below \$50,000.

## 1. Combine these functions to create powerful solutions

### Build comprehensive business analytics tools

	H	I	J	K	L	M	N
1	Commission	Profit Margin	Revenue	Territory Analysis	Growth Rate	12-Month Sales Forecast for Ref 1001	
2	\$188	High	\$116,247	High	0.05	\$3,938	
3	\$115	High	\$117,093	High	0.05	\$4,134	
4	\$90	High	\$65,494	Medium	0.02	\$4,341	
5	\$108	High	\$96,541	Medium	0.09	\$4,558	
6	\$155	High	\$92,108	Medium	0.05	\$4,786	
7	\$120	High	\$81,207	Medium	0.04	\$5,025	
8	\$120	High	\$56,123	Medium	0.05	\$5,277	
9	\$150	High	\$85,903	Medium	0.05	\$5,540	
10	\$105	High	\$57,358	Medium	0.06	\$5,817	
11	\$132	High	\$61,281	Medium	0.03	\$6,108	
12	\$143	High	\$76,190	Medium	0.05	\$6,414	
13	\$165	High	\$54,851	Medium	0.05	\$6,734	
14	\$140	High	\$62,890	Medium	0.05		
15	\$120	High	\$81,207	Medium	0.06		
16	\$198	High	\$90,228	Medium	0.07		
17	\$145	High	\$76,193	Medium	0.05		
18	\$105	High	\$60,846	Medium	0.08		
19	\$300	High	\$97,323	Medium	0.04		
20	\$175	High	\$51,019	Medium	0.04		
21	\$158	High	\$55,767	Medium	0.05		
22	\$162	High	\$74,866	Medium	0.05		
23	\$210	High	\$93,587	Medium	0.03		
24	\$195	High	\$81,388	Medium	0.05		
25	\$185	High	\$63,585	Medium	0.05		
26	\$180	High	\$70,538	Medium	0.05		

When you use SEQUENCE, LET, and LAMBDA together, they solve problems that would otherwise require multiple backing columns or array formulas. This combination creates flexible, maintainable solutions.

Let's look at building a sales forecasting tool using sales data. The following formula calculates a 12-month sales forecast for a single starting sales amount. The formula starts by defining two key variables with LET. The formula takes the value from cell G2 as the initial base sales amount.

=LET(base\_sale, G2, growth\_rate, L2, ProjectMonthly, LAMBDA(month, base\_sale \* (

The formula then takes the monthly growth rate from L2, which is 0.04 (4%). You can change this value to model different scenarios. The formula then defines a small, reusable function called ProjectMonthly. This function calculates the projected sales for a given month based on the base sales and the growth rate.

Furthermore, this formula calls the ProjectMonthly function and passes SEQUENCE(12) into the function. This creates an array of numbers from 1 to 12, and LAMBDA will automatically apply its calculation to each number in that sequence.

This is a real bonus calculator that calculates bonuses based on goal achievement.

=LAMBDA(sales, target, LET(ratio, sales/target, IF(ratio>=1.2, sales\*0.08, IF(ratio

You finished reading the article "**3 Excel Functions That Will Make You a Spreadsheet Expert**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.