

Count cells not between two numbers

Measure	1-Feb	2-Feb	3-Feb	4-Feb	5-Feb	Low	High	Count
A	10	8	11	8	4	5	10	2
B	16	19	19	22	23	15	25	0
C	3	5	3.5	4	2.5	3	5	1
D	0.95	0.99	1.01	1.02	0.85	0	1	2
E	5.5	6.1	10.3	6.7	8.3	5	10	1
F	85	86	88	85	85	85	90	0
G	89	94	91	97	93	90	100	1

Total out of range 7

Generic formula

```
= COUNTIF ( range, "<" & low ) + COUNTIF ( range, ">" & high )
```

Summary

To count cell values that are not between two numbers, you can use the [COUNTIF function](#). In the example shown, the formula in cell K5, copied down, is:

```
= COUNTIF ( C5 : G5, "<" & I5 ) + COUNTIF ( C5 : G5, ">" & J5 )
```

At each new row, this formula returns a count of values *not* between the low and high values in columns I and J.

Explanation

The goal of this example is to count the number values recorded over 5 days that do not fall between two numbers, a low value, and a high value. In other words, to count values that are "out of range". Note that each row, labeled A-G, has its own low and high limit, in columns I and J.

You might at first think to use the [COUNTIFS function](#) with two criteria. However, because COUNTIFS joins criteria with AND logic, it can't be used with two criteria in this scenario. The logic of less than low AND greater than high will always fail, and the result will always be zero. Instead, we need OR logic.

One straightforward solution is to use the [COUNTIF function](#) twice like this:

```
= COUNTIF ( C5 : G5, "<" & I5 ) + COUNTIF ( C5 : G5, ">" & J5 )
```

The first COUNTIF counts values *below* the value in I5, and the second COUNTIF counts values *above* the value in J5. When added together these two results correctly handle the required logic: less than I5 OR greater than J5. Notice the greater than (">") and less than ("<") [operators](#) are [concatenated](#) to cell references with an [ampersand \(&\) operator](#), a quirk of [RACON functions](#).

With SUMPRODUCT

A more elegant solution is to use the [SUMPRODUCT function](#) with two logical expressions:

```
= SUMPRODUCT ( ( C5 : G5 < I5 ) + ( C5 : G5 > J5 ) )
```

Notice we don't need to use concatenation with cell references as with the COUNTIF function above; standard expressions work fine.

This is an example of using [boolean algebra](#) with addition (+), which creates OR logic. When these expressions are evaluated, we have two [arrays](#) of TRUE and FALSE values like this:

```
= SUMPRODUCT ( { FALSE, FALSE, FALSE, FALSE, TRUE } +  
{ FALSE, FALSE, TRUE, FALSE, FALSE } )
```

The math operation automatically coerces the TRUE and FALSE values to 1s and 0s. The result can be visualized like this:

```
= SUMPRODUCT ( { 0, 0, 0, 0, 1 } + { 0, 0, 1, 0, 0 } )
```

This results in a single array containing two 1s:

```
= SUMPRODUCT ( { 0, 0, 1, 0, 1 } )
```

With only one array to process, SUMPRODUCT sums the items in the array and returns a final result of 2.

Conditional formatting

To easily see which values aren't between two values, you can use [a conditional formatting rule with a formula](#).

Measure	1-Feb	2-Feb	3-Feb	4-Feb	5-Feb	Low	High	Count
A	10	8	11	8	4	5	10	2
B	16	19	19	22	23	15	25	0
C	3	5	3.5	4	2.5	3	5	1
D	0.95	0.99	1.01	1.02	0.85	0	1	2
E	5.5	6.1	10.3	6.7	8.3	5	10	1
F	85	86	88	85	85	85	90	0
G	89	94	91	97	93	90	100	1

Total out of range 7

The formula used to highlight the out-of-range values above is:

```
= OR ( C5 < $I5, C5 > $J5 )
```