Data analysis to the MAX()

Enhance your data analysis skills using spreadsheets and data visualization. Increase your productivity and make better business decisions.

TU Delft

edX

Solutions for week 5 of EX101x: Data analysis to the MAX()

Section 5.0: Theory

https://courses.edx.org/courses/DelftX/EX101x/1T2015/courseware/b361de4ea988410aa6203287e356772d/00cb4c193c554820a3dfdff640c545f9/

Questions

1. SO WHAT ARE THE TYPES?

Felienne is looking at the formula LEFT(A1,1) What are the two input types and the output type of this expression?

What is the Input Type of the first argument?

Reference to Text ✔

What is the Input Type of the second argument?

Number ✔

What is the Output Type?

Text/String ✔

A1 is a reference to a string, 1 is a number, and the LEFT function always returns a string. Remember, the left function would also accept a string rather than a reference to a string as a first parameter. Be sure to yell at your computer when Felienne forgets to put the type of the second parameter!

2. SUM()

For this question we consider the SUM(A13:A26).

Here, SUM() operates on a range and returns a single number.

3. MONTH()

Here MONTH() operates on a date and returns a single number.

	A	B
41		
42	13/12/2012	=MONTH(A42)

For this question we consider 13 december 2012.

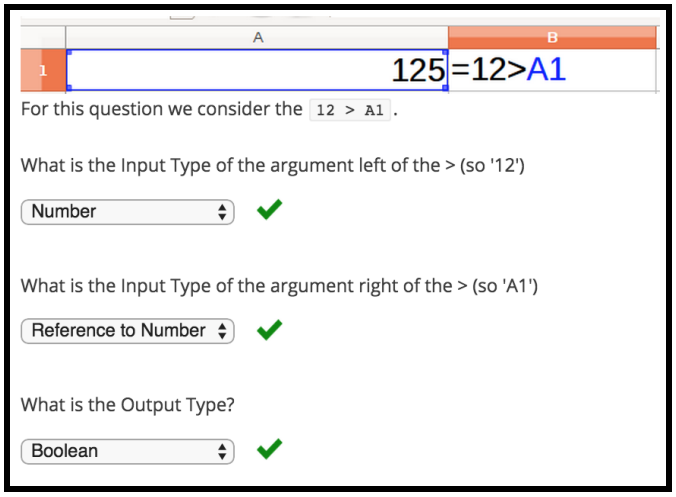
What is the Input Type?

Reference to Date ✔

What is the Output Type?

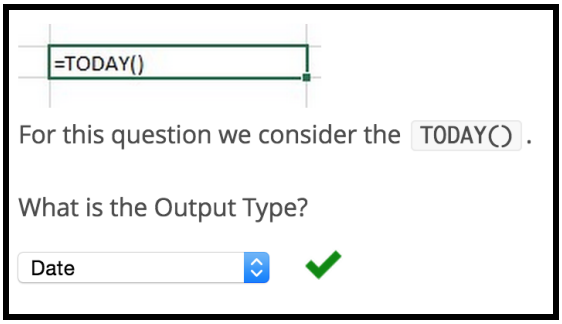
Number ✔

4. GT

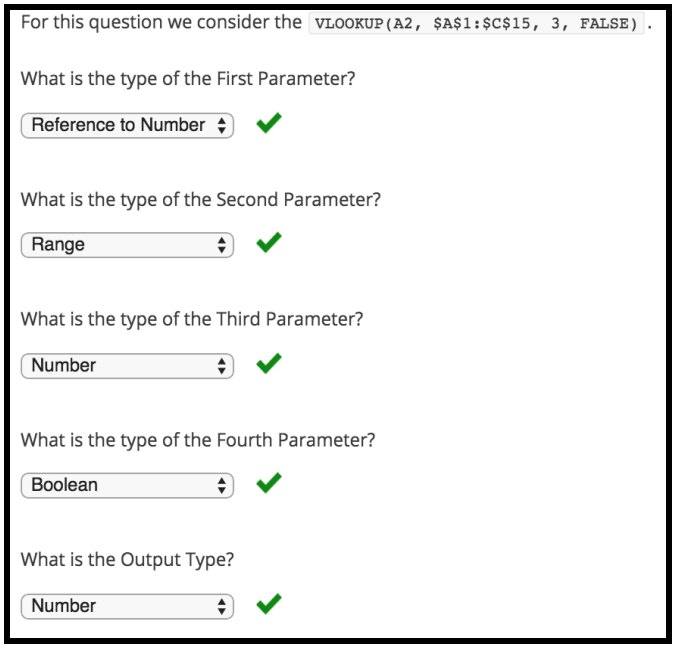


The > takes a number and a reference to a number here and returns TRUE or FALSE (a Boolean).

5. TODAY()



6. VLOOKUP()



We have seen the VLOOKUP() a few times throughout this course now. In this case it operates on a reference to a number, searches in the range, takes the 3rd column (indicated by a number) and the boolean tells us something about the sorting of the data (remember week 2!). It returns the contents of C2, which is simply a year, thus a number (and not a full date).

Section 5.1: Array Basics

https://courses.edx.org/courses/DelftX/EX101x/1T2015/courseware/b361de4ea988410aa6203287e356772d/ec94be0b2c574dcc8f50a459ecb1ab24/

Questions

1. IN VIDEO QUIZ

As you learned, there are different types in excel. A function can take in a certain type(s) and return another. Consider the formula {=AVERAGE(A1:A20-B1:B20)}. What type would this formula return?

While the formula is wrapped in the array function curly brackets, since the outer function is AVERAGE, Excel will perform the inner subtraction on each pair of elements in the range separately, then take the average of all of those numbers.

Section 5.2: Transpose

https://courses.edx.org/courses/DelftX/EX101x/1T2015/courseware/b361de4ea988410aa6203287e356772d/c1af081702ca4bf68a6188e4f968b82a/

Questions

1. WHOOPSY DAISY

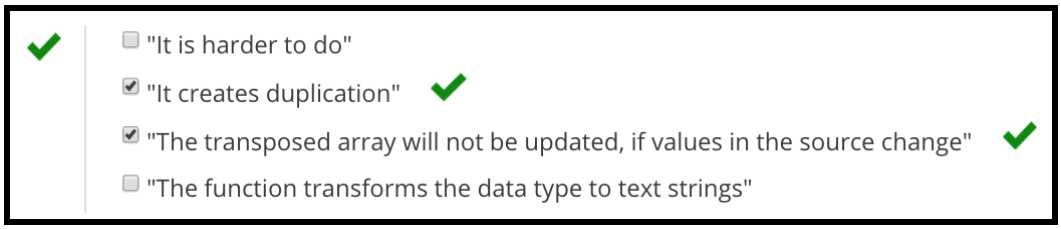
Will the TRANSPOSE function give the correct result when Felienne hits CTRL+SHIFT+ENTER?

No, because although the formula is right, Felienne forgets to give the formula enough space. To see a full explanation, Felienne explains this in the rest of the video.

2. WHAT IF YOU DON'T WANT TO USE THIS METHOD?

There is an alternative too, you could use to transpose the data: right-click, paste special - transpose. If you do not know this function, download the grades sheet here and try it.

What are the downsides of this approach? (2 answers)



It is not harder, actually pasting is easier to perform than copying. But, if your data changes, the copy does not. So indeed, pasting creates duplication. The transposed array will be updated if cell values within the origin changes (however, not if you change the input array size). The function only maps the input data to the transposed array, it will not change the data types.

3. SELECT THE RIGHT TYPE FOR EACH

What type is AVERAGE()?

AVERAGE is a normal function, but can with array arguments and returns a single value, therefore, we call it an Array-behavioral function.

What type is TRANSPOSE()?

TRANSPOSE is an array formula, since it is intended to work on arrays and not on single numbers (transposing one cell is useless)

What type is SUMPRODUCT()?

SUMPRODUCT() is an Array Function since it is designed to have arrays (of the same length) as input and it calculates the inner product (that is mathematical term)

4. THREE MORE

What type is MMULT()?

MMULT() is an array formula since it is designed to work on arrays and not on single cells.

What type is MDETERM()?

MDETERM() is an array formula since it is designed to work on arrays and not on single cells.

What type is LEFT()?

LEFT() is meant to be applied on single values, however it can work in an array as well. But, it cannot return a result in one single cell. So therefore we call it a normal function.

5. ANOTHER!

What type is MINVERSE()?

MINVERSE() is an array function since it is defined to work on an array and not on a single cell.

What type is INDIRECT()?

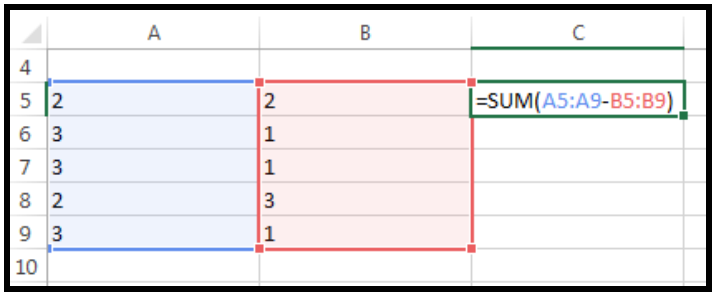
INDIRECT() is normal function, since it is designed to work on single cells. Within an array way of application, the function does not return one single value.

Section 5.4: Assignment

https://courses.edx.org/courses/DelftX/EX101x/1T2015/courseware/b361de4ea988410aa6203287e356772d/c900ad1915744cc59c1756bb0827bb9b/

Questions (Part 1)

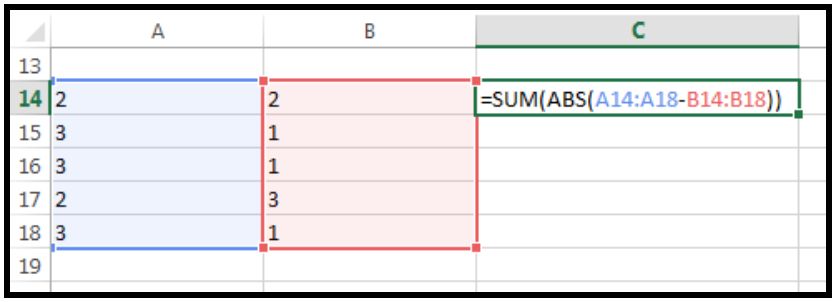
1. SUMMING NUMBERS



What is the result of the array formula in C5?

The answer is 5 since the formula will actually calculate (A5 - B5) + (A6 - B6) + ...

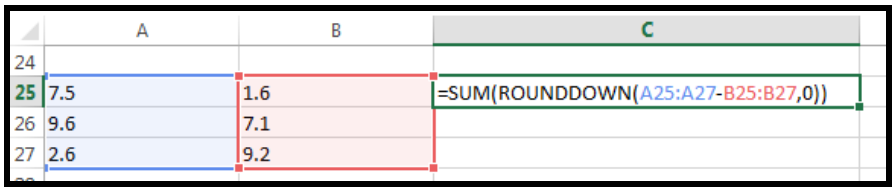
2. SUMMING DIFFERENCES



What is the result of the array formula in C14?

The answer here is 7. The formula sums the absolute differences. Trick is, read the formula from the inside to the outside. So start with the inner arguments. In this case A14:A18-B14:B18. This means that the numbers in column B are subtracted from column A. Row for row. So again A14-B14, A15-B15, … Then the formula take from every outcome the absolute value. So ABS(0,2,2,-1,2) = (0,2,2,1,2). Then the formula sums all individual value so 0+2+2+1+2 = 7. Keep in mind that ABS(2-3) = 1 and not -1.

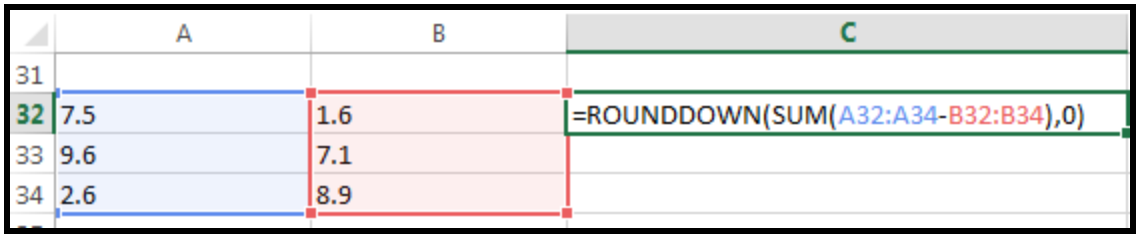
3. ROUNDING DOWN



What is the result of the array formula in C25?

The answer is 1. The formula first calculates the differences between the cells in the columns A and B, then rounds the individual outcomes to the integer value, then takes the sum of all rounded values.

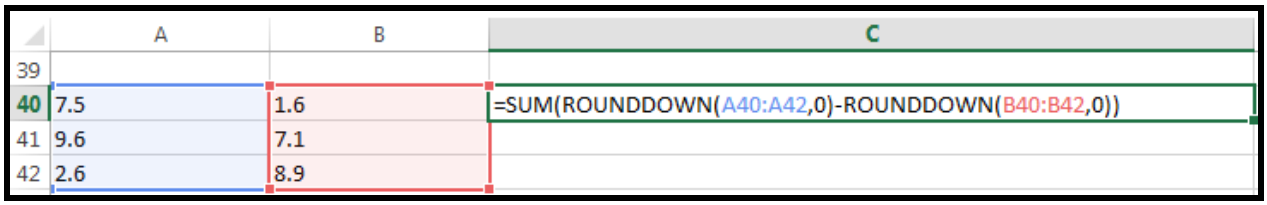
4. ROUNDING DOWN AGAIN?



	A	B	C
31			
32	7.5	1.6	=ROUNDDOWN(SUM(A32:A34-B32:B34),0)
33	9.6	7.1	
34	2.6	8.9	

What is the result of the array formula in C32?
The answer here is 2. The formula first sums the differences between the cells in the columns A and B, then rounds the final outcome to the integer value.

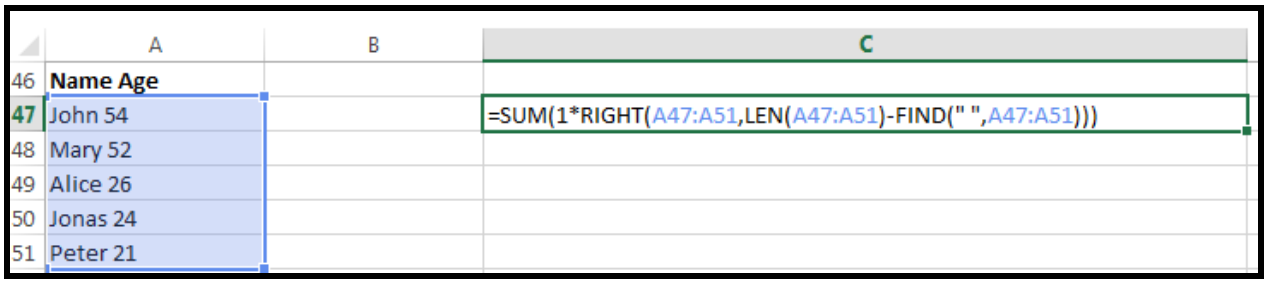
5. ROUNDING DOWN AGAIN AGAIN?



	A	B	C
39			
40	7.5	1.6	=SUM(ROUNDDOWN(A40:A42,0)-ROUNDDOWN(B40:B42,0))
41	9.6	7.1	
42	2.6	8.9	

What is the result of the array formula in C40?
The answer here is also 2. The formula first takes the integer value of each number, then subtracts them from each other and then takes the sum of the differences.

6. A TOUGH ONE!



	A	B	C
46	**Name Age**		
47	John 54		=SUM(1*RIGHT(A47:A51,LEN(A47:A51)-FIND(" ",A47:A51)))
48	Mary 52		
49	Alice 26		
50	Jonas 24		
51	Peter 21		

What is the result of the array formula in C47?
The answer here is 177. This formula shows that you can do multiple things in one array function. So splitting names from ages and then taking the sum of the ages.

Questions (Part 2)

1. LEAST POINTS SCORED

We'll start of relatively simple. What question has contributed the least points in total?

The simplest way to do this is to sum the total number of points for every single question. So SUM(B3:B16) for Q1 etc. The answer is Question 5

F17 =SUM(F3:F16)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	**Question**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**Total**
2	**Perfect**	10	10	10	15	5	5	10	5	10	5	10	5	10
3	Student1	8	10	10	10	5	5	8	5	10	0	10	0	8.1
4	Student2	0	0	0	15	2.5	0	10	5	5	5	10	0	5.25
5	Student3	5	5	10	15	2.5	5	10	5	5	5	3	0	7.05
6	Student4	5	5	10	15	2.5	5	8	5	10	5	10	5	8.55
7	Student5	5	8	10	15	2.5	5	5	5	10	5	10	5	8.55
8	Student6	8	8	10	15	2.5	5	10	5	10	5	10	5	9.35
9	Student7	8	8	0	10	2.5	5	5	0	7	5	5	5	6.05
10	Student8	2	5	10	12	2.5	5	5	0	10	0	10	5	6.65
11	Student9	5	0	5	8	5	5	0	0	7	5	0	5	4.5
12	Student10	0	0	5	5	2.5	5	2	5	5	5	10	5	4.95
13	Student11	0	5	10	5	2.5	0	10	5	10	5	5	5	6.25
14	Student12	5	5	8	10	2.5	5	10	5	10	5	5	5	7.55
15	Student13	5	2	0	2	2.5	0	5	0	5	5	5	5	3.65
16	Student14	0	0	0	15	2.5	5	10	5	5	5	10	5	6.25
17		56	61	88	152	40	55	98	50	109	60	103	55	

2. MOST POINTS LOST

In what question were most points lost?

One way to do this is to use the formula =14*B2 -SUM(B3:B16) for Q1. The maximum number of points is 14 students times the maximum score the specific question. So 14*B2. The number of points achieved equals the sum over the range B3:B16. Subtracting those numbers gives the number of points lost. With this formula you can see that Q1 had the most points lost.

B17 =14*B2 -SUM(B3:B16)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	**Question**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**Total**
2	**Perfect**	10	10	10	15	5	5	10	5	10	5	10	5	10
3	Student1	8	10	10	10	5	5	8	5	10	0	10	0	8.1
4	Student2	0	0	0	15	2.5	0	10	5	5	5	10	0	5.25
5	Student3	5	5	10	15	2.5	5	10	5	5	5	3	0	7.05
6	Student4	5	5	10	15	2.5	5	8	5	10	5	10	5	8.55
7	Student5	5	8	10	15	2.5	5	5	5	10	5	10	5	8.55
8	Student6	8	8	10	15	2.5	5	10	5	10	5	10	5	9.35
9	Student7	8	8	0	10	2.5	5	5	0	7	5	5	5	6.05
10	Student8	2	5	10	12	2.5	5	5	0	10	0	10	5	6.65
11	Student9	5	0	5	8	5	5	0	0	7	5	0	5	4.5
12	Student10	0	0	5	5	2.5	5	2	5	5	5	10	5	4.95
13	Student11	0	5	10	5	2.5	0	10	5	10	5	5	5	6.25
14	Student12	5	5	8	10	2.5	5	10	5	10	5	5	5	7.55
15	Student13	5	2	0	2	2.5	0	5	0	5	5	5	5	3.65
16	Student14	0	0	0	15	2.5	5	10	5	5	5	10	5	6.25
17		84	79	52	58	30	15	42	20	31	10	37	15	

3. WORTH MOST POINTS

What question can contribute most to the final mark, i.e. what question is worth most points?

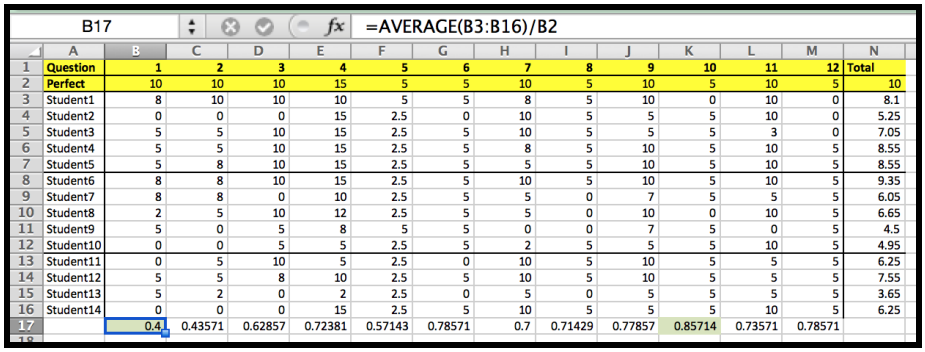
You can easily read this from the maximum obtained scores per question in row 2. The answer is Q4.

4. EASIEST AND HARDEST QUESTIONS

What question was the answered best (relatively)? Question 10

What question was the answered worst (relatively)? Question 1

The relative result per sub question can be calculated with =AVERAGE(B3:B16)/B2 which is the scaled average.



B17 =AVERAGE(B3:B16)/B2

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Question	1	2	3	4	5	6	7	8	9	10	11	12	Total
2	Perfect	10	10	10	15	5	5	10	5	10	5	10	5	10
3	Student1	8	10	10	10	5	5	8	5	10	0	10	0	8.1
4	Student2	0	0	0	15	2.5	0	10	5	5	5	10	0	5.25
5	Student3	5	5	10	15	2.5	5	10	5	5	5	3	0	7.05
6	Student4	5	5	10	15	2.5	5	8	5	10	5	10	5	8.55
7	Student5	5	8	10	15	2.5	5	5	5	10	5	10	5	8.55
8	Student6	8	8	10	15	2.5	5	10	5	10	5	10	5	9.35
9	Student7	8	8	0	10	2.5	5	5	0	7	5	5	5	6.05
10	Student8	2	5	10	12	2.5	5	5	0	10	0	10	5	6.65
11	Student9	5	0	5	8	5	5	0	0	7	5	0	5	4.5
12	Student10	0	0	5	5	2.5	5	2	5	5	5	10	5	4.95
13	Student11	0	5	10	5	2.5	0	10	5	10	5	5	5	6.25
14	Student12	5	5	8	10	2.5	5	10	5	10	5	5	5	7.55
15	Student13	5	2	0	2	2.5	0	5	0	5	5	5	5	3.65
16	Student14	0	0	0	15	2.5	5	10	5	5	5	10	5	6.25
17		0.4	0.43571	0.62857	0.72381	0.57143	0.78571	0.7	0.71429	0.77857	0.85714	0.73571	0.78571	

5. TRY THIS!

*What does this function describe? {=SUM((--(A3:A16="Student1"))*B3:M16)}*

The --(A3:A16 = "Student1") is a logical evaluation. It looks to the range A3:A16 and there where one of the cells within this range equals "Student1" it returns True. The True statement now is converted to a 1 via the --(..) construction, so the double minus sign indicates that the boolean should be converted to a 0 for False or 1 in case of True, dependent on the outcome of the logical evaluation. --(A3:A16="Student1") only returns '1' for cell A3 and '0' for cells A4:A16.

Now the matrix B3:M16 is split up into parts, it is evaluated row-wise. Row B3:M3 is evaluated using the outcome of --(A3="Student1"). Row B4:M4 is evaluated using the outcome of --(A4="Student1"), etc. So --(A3="Student1")=1 is multiplied with every single cell in B3:M3, --(A4="Student1") = 0 is multiplied with every single cell in B4:M4, etc

Finally the sum is applied, so we remain with this:

=SUM(1*B3, 1*C3, 1*D3 … 1*M3, 0*B4, 0*C4, 0*D4, … 0*M4, 0*B5, … 0*M16) = 81
We can also write the array formula with an embedded IF statement.

=SUM((IF(A3:A16="Student1",1,0))*B3:M16). This IF statement is similar to --(A3:A16="Student1")

Given this, we can arrive at the answer shown below.

- ○ It multiplies the student number (1,2,3, ,...) of a specific row with the sum of the points
- ◉ It sums the range B3:M3 since cell A3 contains "Student1" ✔
- ○ It sums all values in B3:M16
- ○ If there is a Student1 in a the range A3:A16 the function sums all values in B3:M16
- ○ The function multiplies strings with numbers and therefore there is no outcome

6. IS PERFECTION REALLY AN IMPOSSIBILITY?

Give the total number of maximal scores which have been obtained over all the questions and all the students. So how often is 10 points obtained if you can earn 10 points maximum, 5 points if you can earn 5 points maximum and 15 if 15 etc. How many perfect scores have been obtained?

Here comes in the nice attributes of array functions. You can really make a very concise formula to answer this question.

First try to make a conditional formula for every subquestion. It reads {=SUM(--(B3:B16=B2))}. Now you have the number of maximum scores per subquestion.

Then, you can think how to generalise this solution. Look at the conditional. It compares the value in rows 3 to 16 in column B with the value in row 2. Maybe we can upscale this, by saying it should compare the values in rows 3:16 with the values in row 2. within a given range.

So try this one: {=SUM(--(B3:M16=B2:M2))}. With this, we can find that the answer is 81.

7. STANDARD DEVIATIONS

What question has the largest standard deviation, i.e. had the most difference in the results?
You can simply apply the formula =STDEV(B3:B16) for Q1, etc. Using this, you can find that the answer is Q4.

E17 =STDEV(E3:E16)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	**Question**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**Total**
2	**Perfect**	10	10	10	15	5	5	10	5	10	5	10	5	10
3	Student1	8	10	10	10	5	5	8	5	10	0	10	0	8.1
4	Student2	0	0	0	15	2.5	0	10	5	5	5	10	0	5.25
5	Student3	5	5	10	15	2.5	5	10	5	5	5	3	0	7.05
6	Student4	5	5	10	15	2.5	5	8	5	10	5	10	5	8.55
7	Student5	5	8	10	15	2.5	5	5	5	10	5	10	5	8.55
8	Student6	8	8	10	15	2.5	5	10	5	10	5	10	5	9.35
9	Student7	8	8	0	10	2.5	5	5	0	7	5	5	5	6.05
10	Student8	2	5	10	12	2.5	5	5	0	10	0	10	5	6.65
11	Student9	5	0	5	8	5	5	0	0	7	5	0	5	4.5
12	Student10	0	0	5	5	2.5	5	2	5	5	5	10	5	4.95
13	Student11	0	5	10	5	2.5	0	10	5	10	5	5	5	6.25
14	Student12	5	5	8	10	2.5	5	10	5	10	5	5	5	7.55
15	Student13	5	2	0	2	2.5	0	5	0	5	5	5	5	3.65
16	Student14	0	0	0	15	2.5	5	10	5	5	5	10	5	6.25
17		3.06343	3.45537	4.4795	4.48686	0.90784	2.12908	3.35123	2.34404	2.39161	1.81568	3.41055	2.12908	
18														

8. DOES IT WORK?

*The ultimate goal of the teacher is to make the standard deviations between the questions more or less equal. Therefore, the professor re-do his analysis. He filters out all students with a final grade <= 4.5 and >=9.0. Then, he determines the STDEV of Question 1. To do so, he uses the following formula: {=STDEV(--($N$3:$N$16>4.5)*--($N$3:$N$16<9)*B3:B16)}.*

A smart student has gotten hold of this formula used by the teacher. What is his reaction to it?

The solution is less trivial than one might think of. The trivial way is to apply two conditionals on the final mark. One to leave out marks <=4.5 and one to leave out marks >=9.0. Followed by taking the STDEV of the result.

That reads like {=STDEV(--($N$3:$N$16>4.5)*--($N$3:$N$16<9)*B3:B16)} for question 1.

However, we now really have to dive into the code. As said, the conditionals returns 1 and 0 for True and False. Multiplying with the values in Range B3:B16. However, it will NOT FILTER OUT the numbers, it replaces them with 0. And therefore the outcome is not correct.

- ○ The formula calculates the correct 'new' stdev for question 1. So the formula is correct.
- ○ The formula could work, but there is a minor typo (typing mistake) in the formula.
- ○ The formula could work, but the wrong values are filtered out.
- ○ The formula could work, but the standard deviation is taken over the wrong range.
- ◉ The formula can not work, as the values are not filtered, but replaced by zero. ✔
- ○ The formula can not work, as one should use another function instead of STDEV.

9. THE AVERAGE SCORE

Give the average final grade over all students on this test. Round your answer to one decimal place.

You can simply take the average over the final marks. So =AVERAGE(N3:N16). Using this, you can find that the average is 6.6.

10. NUMBER OF PASSING STUDENTS

This professor is a little picky, and he wants that only 50% of the students passes this course at the first attempt. All students having a 5.5 or higher pass the test. Try to see if you can do this with both a normal formula and an array formula.
How many students currently pass the test?

With a normal formula, you can use =COUNTIF(N316,">="&5.5)

The array way of doing this, reads {=SUM(--(N3:N16>=5.5))}

The answer for either strategy is 10.

11. ADJUSTING THE SCORES

In the professor's opinion, too many people are passing the test. There are several options to adjust the grade. He decides to leave out questions 6 and 10 in the analysis, since those questions are made best. In order to still have a 10 as final result, he multiplies the other credits with 1.111. What is the new mark of Student7, rounded to one decimal?

The answer reads as follows:
{=SUM(B3:M16*--($B$1:$M$1<>6)*--($B$1:$M$1<>10)*--(A3:A16="Student7"))/10*1.111}.

Actually, we calculate a new weighted average for all students, and we only consider student7 without question 6 and 10.

The sumrange is B3:M16.

Filter for question 6: --($B$1:$M$1<>6)

Filter for question 10: --($B$1:$M$1<>10)

Filter for Student7: --(A3:A16="Student7")

Then we have to divide it by 10 to come to from a maximum score of 100 to 10.

Finally we multiply the outcome with 1.111 to compensate for the points from question 6 and 10 (as stated in the assignment).

Using this, we can find that the answer is 5.6

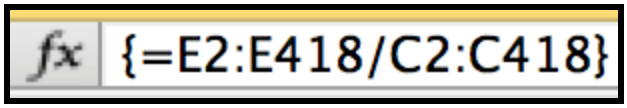
Section 5.5: Bonus Exercise

https://courses.edx.org/courses/DelftX/EX101x/1T2015/courseware/b361de4ea988410aa6203287e356772d/0d3cd8e3c9ee4a0c9f454690ca33d86f/

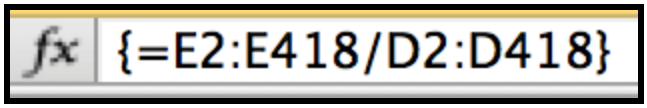
There were four parts to this exercise, we will explain each part separately below.

Part 1: Add Metrics to the Transaction Sheet

The first metric we want to add to the transaction sheet is the cash per square foot metric. In the spirit of this week, we will do this with an array formula. We can use the formula below to get the cash per square foot in each cell.



The second metric is the percent of list price. We can also use an array formula to find this.



Part 2: Add Metrics to the Price Sheet

The first metric we want to calculate is the average price per square foot. We first name the Zip Code column in the transaction sheet as “ZipCodes” and the Sq. Ft. column in the transaction sheet as “SqFt”. Then, we can apply the following formula to find this metric:

```
=SUMIF(ZipCodes,A2,CashPSF) / SUMIF(ZipCodes,A2,SqFt)
```

The second metric we want to calculate is the target price. We can use the same ZipCodes named range as before and apply the following formula to find target price.

```
=A2 * SUMIF(ZipCodes,A2,SqFt) / COUNTIF(ZipCodes,A2)
```

Part 3: Number of Deals per Zip

For this task, there are two simple ways of solving it. First, we can use a pivottable on the transactions sheet. We use the zip codes as rows and a count of deals as the values.

Alternatively, we can use a simple COUNTIF function in the deals_by_zip sheet “Count of Deals” column. The formula for row 2 is shown below.

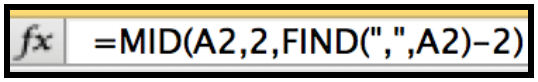
```
=COUNTIF(transactions!B:B,A2)
```

With either method, we get the values shown below.

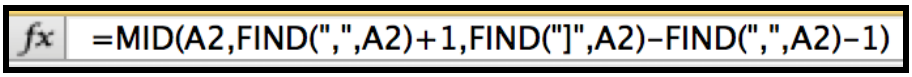
	A	B
1	Zip code	Count of deals
2	90195	27
3	46800	26
4	84559	25
5	79726	25
6	54500	25
7	68399	23
8	87812	22
9	64126	21
10	13878	21
11	99234	21
12	55179	21
13	92192	20
14	87476	20
15	22177	19
16	34243	19
17	29348	18
18	61724	17
19	67199	16
20	10733	16
21	13313	15

Part 4: Number of Deals per Price Range

The first challenge of this metric is to pull out the minimum and maximum values of the given price range. For this, we will add two new columns to the sorted_by_priceRange sheet, “Min” and “Max”. We use the formula



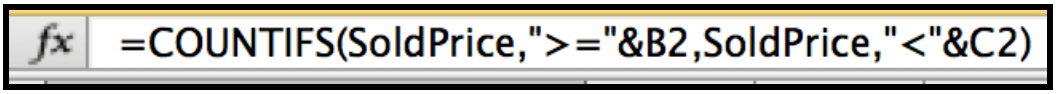
to find the min value based off of the range given. Then we use the formula



to find the maximum value of the range given. If you apply these two you get the table shown below.

	A	B	C
1	**Price range**	**Min**	**Max**
2	[80001,90000]	80001	90000
3	[30001,40000]	30001	40000
4	[20001,30000]	20001	30000
5	[70001,80000]	70001	80000
6	[50001,60000]	50001	60000
7	[10001,20000]	10001	20000
8	[60001,70000]	60001	70000
9	[90001,100000]	90001	100000
10	[40001,50000]	40001	50000
11	[0,10000]	0	10000

Now we can use these two extracted values to find how many deals fall into this range. The formula below does the trick:



Using this, we get the result shown below!

	A	B	C	D
1	**Price range**	**Min**	**Max**	**Count of deals**
2	[80001,90000]	80001	90000	55
3	[30001,40000]	30001	40000	52
4	[20001,30000]	20001	30000	49
5	[70001,80000]	70001	80000	49
6	[50001,60000]	50001	60000	48
7	[10001,20000]	10001	20000	47
8	[60001,70000]	60001	70000	40
9	[90001,100000]	90001	100000	39
10	[40001,50000]	40001	50000	38
11	[0,10000]	0	10000	0