Package 'ExcelFunctionsR'

July 21, 2025

Type Package

Title Imports Excel Functions to R

Version 0.1.4

Author Irakli Salia <irakli.salia854@gmail.com>

Maintainer Nika Salia <2001salia@gmail.com>

Description

Implements 'Excel' functions in 'R' for your calculation simplicity.You can use most of the aggregate functions, addressing functions,logical functions and text functions. Helps you a ton in learning how 'R' works as some 'Excel' users might be struggling with the program.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Imports stringr, tidyr, lubridate, roperators, plyr, stats

Depends R (>= 2.10)

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2020-06-22 09:40:03 UTC

Contents

AND *Basic AND function from excel*

Description

It acts similiarly to Excel's AND function. You give the function logical arguments and it either returns true or false.

Usage

```
AND(
  logical1,
  logical2 = TRUE,
  logical3 = TRUE,
  logical4 = TRUE,
  logical5 = TRUE,
  logical6 = TRUE,
  logical7 = TRUE,
  logical8 = TRUE,
  logical9 = TRUE,
  logical10 = TRUE,
  logical11 = TRUE,
  logical12 = TRUE,
  logical13 = TRUE,
  logical14 = TRUE,
  logical15 = TRUE,
  logical16 = TRUE,
  logical17 = TRUE,
  logical18 = TRUE,
  logical19 = TRUE,
  logical20 = TRUE,
  logical21 = TRUE,
  logical22 = TRUE,
  logical23 = TRUE,
  logical24 = TRUE,
  logical25 = TRUE,
  logical26 = TRUE,
  logical27 = TRUE,
  logical28 = TRUE,
  logical29 = TRUE,
  logical30 = TRUE,
  logical31 = TRUE,
  logical32 = TRUE
)
```

Arguments

`logical1, logical2, logical3, logical4, logical5, logical6, logical7, logical8, logical9, logical10, logical11, logical12, logical13, logical14, logical15, logical16, logical17, logical18, logical19, logical20, logical21, logical22, logical23, logical24, logical25, logical26, logical27, logical28, logical29, logical30, logical31, logical32`

Specify logicals as arguments. The function follows OR logic.

Value

In the example we take a built-in dataset called iris and see which species are called setosa and which one has a petal length of 1.4. If both of these conditions return true then the final answer will also be true. Function will always return a logical class.

Examples

```
AND(iris$Species == "setosa", iris$Petal.Length == 1.4)
```

AVERAGE *Basic AVERAGE function from excel*

Description

It acts similiarly to Excel's AVERAGE function. It simply calculates average of the given numbers.

Usage

```
AVERAGE(
  number1,
  number2 = NA,
  number3 = NA,
  number4 = NA,
  number5 = NA,
  number6 = NA,
  number7 = NA,
  number8 = NA,
  number9 = NA,
  number10 = NA,
  number11 = NA,
  number12 = NA,
  number13 = NA,
  number14 = NA,
  number15 = NA,
  number16 = NA,
  number17 = NA,
  number18 = NA,
```

```
  number19 = NA,
  number20 = NA,
  number21 = NA,
  number22 = NA,
  number23 = NA,
  number24 = NA
)
```

Arguments

number1, number2, number3, number4, number5, number6, number7, number8, number9, number10, number11, number12, number13, number14, number15, number16, number17, number18, number19, number20, number21, number22, number23, number24

Give this function number, same goes for other number arguments as well, but they are optional.

Value

As you can see in the example below, the average of numbers 10,20,30,40 is 25. By default excel removes NA values, by NA values I mean Excel's blank cells. Function will always return numeric class.

Examples

```
AVERAGE(10,20,30,40)
```

AVERAGEIF *Basic AVERAGEIF function from excel*

Description

It acts similiarly to Excel's AVERAGEIF function. It calculates the average of the values where certain criterias are met.

Usage

```
AVERAGEIF(range, criteria, average_range)
```

Arguments

range	Give this function argument range for it to evaluate your criteria.
criteria	Give this function a criteria so it can check the range for this criteria.
average_range	Give this function a range for it to average on. So first it evaluates range argument based on criteria and it averages the numbers that meet the criteria.

Value

It takes the average of the column data where there are certain conditions met. In the example you can see we are testing if Species equal setosa and wherever this holds true we average the numbers. Example's result show the average of the Petal width of setosa Species. Function will always return numeric class.

Examples

```
AVERAGEIF(iris$Species,"setosa",iris$Petal.Width)
```

AVERAGEIFS *Basic AVERAGEIFS function from excel*

Description

It acts similiarly to Excel's AVERAGEIFS function. It calculates the average of the values where several criterias are met(it mimics and expression for criterias).

Usage

```
AVERAGEIFS(
  average_range,
  criteria_range1,
  criteria1,
  criteria_range2 = TRUE,
  criteria2 = TRUE,
  criteria_range3 = TRUE,
  criteria3 = TRUE,
  criteria_range4 = TRUE,
  criteria4 = TRUE,
  criteria_range5 = TRUE,
  criteria5 = TRUE,
  criteria_range6 = TRUE,
  criteria6 = TRUE,
  criteria_range7 = TRUE,
  criteria7 = TRUE,
  criteria_range8 = TRUE,
  criteria8 = TRUE,
  criteria_range9 = TRUE,
  criteria9 = TRUE,
  criteria_range10 = TRUE,
  criteria10 = TRUE
)
```

Arguments

`average_range`	Give this function a range for it to average on. So first it evaluates range argument based on criteria and it averages the numbers that meet the criteria.
`criteria_range1`, `criteria_range2`, `criteria_range3`, `criteria_range4`, `criteria_range5`, `criteria_range6`, `criteria_range7`, `criteria_range8`, `criteria_range9`, `criteria_range10`	Give this function a criteria_range/ranges so it can check the range for the appropriate criteria. criteria_range1 is checked against criteria1
`criteria1`, `criteria2`, `criteria3`, `criteria4`, `criteria5`, `criteria6`, `criteria7`, `criteria8`, `criteria9`, `criteria10`	Give this function a criteria so it can check the appropriate criteria_range for it. criteria1 for criteria_range1

Value

In this example we average Sepal Width of virginica species who have petal width less than 2. Function will always return numeric class.

Examples

```
AVERAGEIFS(iris$Sepal.Width,iris$Species,"virginica",iris$Petal.Width,"<2")
```

CONCAT *Basic Concat function from excel*

Description

It acts similiarly to Excel's CONCAT function. It concatenates given strings together, it can concatenates maximum 32 values.

Usage

```
CONCAT(
  text1,
  text2,
  text3 = "",
  text4 = "",
  text5 = "",
  text6 = "",
  text7 = "",
  text8 = "",
  text9 = "",
  text10 = "",
  text11 = "",
  text12 = "",
  text13 = "",
```

```
  text14 = "",
  text15 = "",
  text16 = "",
  text17 = "",
  text18 = "",
  text19 = "",
  text20 = "",
  text21 = "",
  text22 = "",
  text23 = "",
  text24 = "",
  text25 = "",
  text26 = "",
  text27 = "",
  text28 = "",
  text29 = "",
  text30 = "",
  text31 = "",
  text32 = ""
)
```

Arguments

`text1`, `text2`, `text3`, `text4`, `text5`, `text6`, `text7`, `text8`, `text9`, `text10`, `text11`, `text12`, `text13`, `text14`, `text15`, `text16`, `text17`, `text18`, `text19`, `text20`, `text21`, `text22`, `text23`, `text24`, `text25`, `text26`, `text27`, `text28`, `text29`, `text30`, `text31`, `text32`

Give this function the text to concatenate.text1 and text2 arguments are mandatory, while others are optional.

Value

In the example we can see the string. We had two different strings and after concatenating we get them together. This function will always return string class.(Character in Excel language).

Examples

```
CONCAT("Concatenate this ","to this")
```

CONCATENATE *Basic Concatenate function from excel*

Description

It acts similiarly to Excel's CONCATENATE function. Same as the CONCAT function but for users of old Excel the version to concatenate strings is CONCATENATE so I include it in this package.

Usage

```
CONCATENATE(
  text1,
  text2,
  text3 = "",
  text4 = "",
  text5 = "",
  text6 = "",
  text7 = "",
  text8 = "",
  text9 = "",
  text10 = "",
  text11 = "",
  text12 = "",
  text13 = "",
  text14 = "",
  text15 = "",
  text16 = "",
  text17 = "",
  text18 = "",
  text19 = "",
  text20 = "",
  text21 = "",
  text22 = "",
  text23 = "",
  text24 = "",
  text25 = "",
  text26 = "",
  text27 = "",
  text28 = "",
  text29 = "",
  text30 = "",
  text31 = "",
  text32 = ""
)
```

Arguments

`text1`, `text2`, `text3`, `text4`, `text5`, `text6`, `text7`, `text8`, `text9`, `text10`, `text11`, `text12`, `text13`, `text14`, `text15`, `text16`, `text17`, `text18`, `text19`, `text20`, `text21`, `text22`, `text23`, `text24`, `text25`, `text26`, `text27`, `text28`, `text29`, `text30`, `text31`, `text32`

Give this function the text to concatenate.text1 and text2 arguments are mandatory, while others are optional.

Value

In the example we can see the string. We had two different strings and after concatenating we get them together. This function will always return string class(Character in Excel language).

Examples

```
CONCATENATE("Thanks to GM"," for datacamp")
```

`COUNT` *Basic COUNT function from excel*

Description

It acts similiarly to Excel's COUNT function. It counts the amount of values in the given array.

Usage

```
COUNT(value)
```

Arguments

`value`	Count amount of the values in the range.

Value

In this example we count the amount of species in the built-in iris dataset. Function will always return numeric class.

Examples

```
COUNT(iris$Species)
```

`COUNTIF` *Basic COUNTIF function from excel*

Description

It acts similiarly to Excel's COUNTIF function. It counts the amount of cells that comply with the given criteria.

Usage

```
COUNTIF(range, criteria)
```

Arguments

`range`	Specify range for Countif
`criteria`	Give the criteria to check the range for.

Value

In this example we count the amount of setosa in iris dataset. Function will always return numeric class.

Examples

```
COUNTIF(iris$Species,"setosa")
```

COUNTIFS *Basic COUNTIFS function from excel*

Description

It acts similiarly to Excel's COUNTIFS function. Counts values in a range which comply with given criteria.

Usage

```
COUNTIFS(
  criteria_range1,
  criteria1,
  criteria_range2 = TRUE,
  criteria2 = TRUE,
  criteria_range3 = TRUE,
  criteria3 = TRUE,
  criteria_range4 = TRUE,
  criteria4 = TRUE,
  criteria_range5 = TRUE,
  criteria5 = TRUE,
  criteria_range6 = TRUE,
  criteria6 = TRUE,
  criteria_range7 = TRUE,
  criteria7 = TRUE,
  criteria_range8 = TRUE,
  criteria8 = TRUE,
  criteria_range9 = TRUE,
  criteria9 = TRUE,
  criteria_range10 = TRUE,
  criteria10 = TRUE
)
```

Arguments

`criteria_range1, criteria_range2, criteria_range3, criteria_range4, criteria_range5, criteria_range6, criteria_range7, criteria_range8, criteria_range9, criteria_range10`

Specify range for Countifs, only criteria_range1 is mandatory.

`criteria1, criteria2, criteria3, criteria4, criteria5, criteria6, criteria7, criteria8, criteria9, criteria10`
: Give the criteria to check the range for.Only criteria1 is necessary, others are optional.

Value

In this example we count the amount of cells where Spcies are setosa and has a Petal Width of 0.2 .Function will always return numeric class.

Examples

```
COUNTIFS(iris$Species,"setosa",iris$Petal.Width,0.2)
```

DATE *Basic DATE function from excel*

Description

It acts similiarly to Excel's DATE function. You give 3 arguments which are year,month and day and it will give you the date in a date format.

Usage

```
DATE(year, month, day)
```

Arguments

`year`	Give year argument to the function.
`month`	Give month argument to the function.
`day`	Give day argument to the function.

Value

This example returns 23rd June of 2020. Function will always return Date class.

Examples

```
DATE(2020,23,06)
```

DATEDIF *Basic DATEDIF function from excel*

Description

It acts similiarly to Excel's DATEDIF function. It returns difference between two dates, either day,month or year, it's up to the user to specify which type of difference user wants.

Usage

```
DATEDIF(start_date, end_date, difference = "d")
```

Arguments

start_date	Start date to evaluate the difference
end_date	End Date to evaluate the difference
difference	What type of difference do you want? Year,Month or Day? Specify "m" for example for month/months, "d" for day/days and "y" for year/years.

Value

In these examples we have all 3 types of returns, first is difference between specified two dates in days, second one is difference in months and third one is difference in years.Function will always return numeric class.

Examples

```
DATEDIF(DATE(2020,1,1),DATE(2020,2,1),"d")
DATEDIF(DATE(2020,1,1),DATE(2020,2,1),"m")
DATEDIF(DATE(2020,1,1),DATE(2020,2,1),"Y")
```

DATETOEXCELSERIES *Date to excel date series function*

Description

Functions converts dates to Excel General date series which might be useful when writing Excel files.

Usage

```
DATETOEXCELSERIES(date)
```

Arguments

date	Convert R date type to Excel general date series, this might be helpful for Excel users.

Value

In this example it returns the Excel's general date series equivalent of date 1st January of 2020. Function will always return numeric class.

Examples

```
DATETOEXCELSERIES(DATE(2020,1,1))
```

DAY *Basic DAY function from excel*

Description

It acts similiarly to Excel's DAY function. It gives you the day from specified date.

Usage

```
DAY(date)
```

Arguments

date	Give the date argument so it can extract day from the date.

Value

in this example we have 13th of January. Function will return 13 as it is the day from the date. Function will always return numeric class.

Examples

```
DAY(DATE(2020,1,13))
```

DAYS *Basic DAYS function from excel*

Description

It acts similiarly to Excel's DAYS function. It calculates the difference between two dates in days.

Usage

```
DAYS(start_date, end_date)
```

Arguments

start_date	Give the start_date argument so it can calculate days.
end_date	Give the end_date argument so it can calculate days.

Value

In this example we are interested how many days there are between 1st February 2020 and 15th February 2020 which is 14. Function will always return numeric class.

Examples

```
DAYS(DATE(2020,2,1),DATE(2020,2,15))
```

EOMONTH *Basic EOMONTH function from excel*

Description

It acts similiarly to Excel's EOMONTH function. It returns the end of month date for the specified date.

Usage

```
EOMONTH(date, months = 0)
```

Arguments

date	Give the date argument so it can give you the end of the month.
months	The number of months before or after start_date. A positive value for months yields a future date; a negative value yields a past date.

Value

In this case we specify 2nd June 2008. Function returns end of the month which is 30th June 2008. Function will always return Date class.

Examples

```
EOMONTH(DATE(2008,6,2))
```

FIND *Basic FIND function from excel*

Description

It acts similiarly to Excel's FIND function. It finds the starting point of the string where it matches your find_text value.

Usage

```
FIND(find_text, within_text)
```

Arguments

find_text	Find the text in the text.
within_text	Where should the function find the text.

Value

in this example we try to find on which place does CRAN start. Function will always return numeric class.

Examples

```
FIND("CRAN","I LOVE CRAN")
```

IF *Basic If function from excel*

Description

It acts similiarly to Excel's If function. Works on vectors as well. IF function is one of the first logical functions which has 3 arguments, logical test, value if true and value if false. If logical test passes(meaning it returns true) then function goes to value if true, otherwise it goes to value if false argument.

Usage

```
IF(logical_test, valueifTrue = 0, valueifFalse = 0)
```

Arguments

logical_test	This is the usual test we run in excel which returns either TRUE or FALSE value. Use double equal signs for logical test if you want to equal.
valueifTrue	If the logical_test evaluates to TRUE then function will return the value you input here
valueifFalse	If the logical_test evaluates to FALSE then function will return the value you input here

Value

In this example we test if Species equal virginica and if it does we get a return Yes, otherwise it returns No. Function can return different classes, it depends on what you specify in value if true and what you specify in value if false.

Examples

```
IF(iris$Species == "virginica","Yes","No")
```

`IFNA` *Basic IFNA function from excel*

Description

It acts similiarly to Excel's IFNA function. If value is NA(or blank in Excel terms) then the function will return the second argument, if not then it will return the non-NA value which is the first argument.

Usage

```
IFNA(value, value_if_na)
```

Arguments

`value`	Evaluate if it is NA.
`value_if_na`	What should the function do if the value is NA.

Value

In this case the function returns "It is NA" as we specify the first value NA. Function can return different classes because first argument can be either character,numeric, logical or anything else.

Examples

```
IFNA(NA,"It is NA")
```

INDEX *Basic INDEX function from excel*

Description

It acts similiarly to Excel's INDEX function. It gives you the value from dataframe when you specify the array indices(row and column)

Usage

```
INDEX(array, row_num, column_num = 1)
```

Arguments

array	Which array/table should it use?
row_num	Which row should it return the value from?
column_num	Which column should it return the value from?

Value

In this example we get 3rd row and 2nd column from the dataframe. This function can return different classes numeric, character, logical etc. It depends on what is in array/dataframe.

Examples

```
INDEX(iris,3,2)
```

ISBLANK *Basic ISBLANK function from excel*

Description

It acts similiarly to Excel's ISBLANK function. If the value you give is blank(NA in R terms) then it returns true, in other cases it returns false.

Usage

```
ISBLANK(value)
```

Arguments

value	Give the function the value for it to evaluate if it is blank?In R words if it is NA. NA is blank in R.

Value

Function returns logical class. If the value specified is blank then it returns true, in all other cases it returns false. Function will always return logical class.

Examples

```
ISBLANK(NA)
ISBLANK(212)
ISBLANK("asdasd")
ISBLANK(iris$Species)
```

ISEVEN *Basic ISEVEN function from excel*

Description

It acts similiarly to Excel's ISEVEN function. If the specified number is even then it returns true, if not then false.

Usage

```
ISEVEN(number)
```

Arguments

number	Input the number for it to evaluate if it is even?

Value

First example returns true as it is an even number 2, second example returns false as it isn't an even number. Function will always return logical class.

Examples

```
ISEVEN(2)
ISEVEN(1)
```

ISLOGICAL *Basic ISLOGICAL function from excel*

Description

It acts similiarly to Excel's ISLOGICAL function. If specified value is true or false then it returns true, if not then it returns false.

Usage

```
ISLOGICAL(value)
```

Arguments

value	Input the number for it to evaluate if it is logical? Works on vectors/arrays as well.

Value

We have 3 cases in the examples. First one is logical therefore function returns true, second one is also logical and it returns true as well. Third example isn't logical therefore function returns false. Function will always return logical class.

Examples

```
ISLOGICAL(TRUE)
ISLOGICAL(FALSE)
ISLOGICAL("Is this a logical?")
```

ISNUMBER *Basic ISNUMBER function from excel*

Description

It acts similiarly to Excel's ISNUMBER function. If the specified value is a number it returns true, in all other cases it returns false.

Usage

```
ISNUMBER(value)
```

Arguments

value	Input the number for it to evaluate if it is number? Works on vectors/arrays as well.

Value

first example returns true as it is a number. Second example returns false as it isn't a number, it's a string. Function will always return logical class.

Examples

```
ISNUMBER(2)
ISNUMBER("2")
```

`ISODD` *Basic ISODD function from excel*

Description

It acts similiarly to Excel's ISODD function. If the specified number is odd then it returns true, if not then false.

Usage

```
ISODD(number)
```

Arguments

`number`	Input the number for it to evaluate if it is an odd number? Works on vectors/arrays as well.

Value

First example returns true as it is an odd number 1, second example returns false as it isn't an odd number. Function will always return logical class.

Examples

```
ISODD(1)
ISODD(2)
```

LEFT *Basic LEFT function from excel*

Description

It acts similiarly to Excel's LEFT function. It takes the text and gives you the amount of characters you want to get from the string.

Usage

```
LEFT(text, num_chars)
```

Arguments

text	the text you want to select characters from left.
num_chars	How many characters should it select?

Value

In this case we have a sentence and we want to extract first 4 characters from the sentence. Therefore we specify the argument 4 and it gives us the first word.Function will always return character class.

Examples

```
LEFT("Fear what happens",4)
```

LEN *Basic LEN function from excel*

Description

It acts similiarly to Excel's LEN function. This function gives you the length of a string.

Usage

```
LEN(text)
```

Arguments

text	amount of characters in the word.

Value

in this example we see how long the sentence is. Function will always return numeric class.

Examples

```
LEN("This is great!")
```

LOWER *Basic LOWER function from excel*

Description

It acts similiarly to Excel's LOWER function. It converts the sentence/word to lowercase characters.

Usage

```
LOWER(text)
```

Arguments

text	Give the function a word to make it lower.Give the texts via vector if you want to perform it on multiple texts.

Value

In this case we lower the whole specified sentence and return the sentence in all lower characters. Function will always return character class.

Examples

```
LOWER("THIS IS SPARTAA! IS IT THOUGH AFTER LOWERING?")
```

MATCH *Basic MATCH function from excel*

Description

It acts similiarly to Excel's MATCH function. It matches the value in the array.

Usage

```
MATCH(lookup_value, lookup_array)
```

Arguments

lookup_value	what value to lookup
lookup_array	Where should it lookup the value

Value

This example gives us the first index of an array where Species is virginica. Function will always return numeric.

Examples

```
MATCH("virginica",iris$Species)
```

`MAXIF` *Basic MAXIF function from excel*

Description

It acts similiarly to Excel's MAXIF function. It returns the maximum value from an array after testing for certain criterias.

Usage

```
MAXIF(range, criteria, max_range)
```

Arguments

`range`	Range where it should check the criteria
`criteria`	Where should it lookup the value
`max_range`	Which array should it return the max from.

Value

In this case we get the maximum value of Sepal Length from Species which are virginica. Function will always return numeric class.

Examples

```
MAXIF(iris$Species,"virginica",iris$Sepal.Length)
```

`MAXIFS` *Basic MAXIFS function from excel*

Description

It acts similiarly to Excel's MAXIFS function. It returns the maximum value from an array after testing for several criterias.

Usage

```
MAXIFS(
  max_range,
  criteria_range1,
  criteria1,
  criteria_range2 = TRUE,
  criteria2 = TRUE,
  criteria_range3 = TRUE,
  criteria3 = TRUE,
  criteria_range4 = TRUE,
  criteria4 = TRUE,
  criteria_range5 = TRUE,
  criteria5 = TRUE,
  criteria_range6 = TRUE,
  criteria6 = TRUE,
  criteria_range7 = TRUE,
  criteria7 = TRUE,
  criteria_range8 = TRUE,
  criteria8 = TRUE,
  criteria_range9 = TRUE,
  criteria9 = TRUE,
  criteria_range10 = TRUE,
  criteria10 = TRUE
)
```

Arguments

`max_range`	Range from where it should return the maximum value from.
`criteria_range1`, `criteria_range2`, `criteria_range3`, `criteria_range4`, `criteria_range5`, `criteria_range6`, `criteria_range7`, `criteria_range8`, `criteria_range9`, `criteria_range10`	Which range should the criteria tested for.Only criteria_range1 is mandatory,others are optional.
`criteria1`, `criteria2`, `criteria3`, `criteria4`, `criteria5`, `criteria6`, `criteria7`, `criteria8`, `criteria9`, `criteria10`	What criteria should the range be checked against. Only criteria1 is mandatory, others are optional.

Value

This example returns maximum value of Petal Length of species setosa who have petal width 0.2. Function will always return numeric class.

Examples

```
MAXIFS(iris$Petal.Length,iris$Species,"setosa",iris$Petal.Width,0.2)
```

MID *Basic MID function from excel*

Description

It acts similiarly to Excel's MID function. Function is for string extraction. You select the starting number and the amount of characters you want to extract.

Usage

```
MID(text, start_num, num_chars)
```

Arguments

text	From whch text should it return the string?
start_num	Where should it start counting from?
num_chars	How many characters should it return?

Value

In this example we want to extract "kata" from this string. So we specify 5 as the starting number and 4 as the amount of characters to extract. Function will always return character class.

Examples

```
MID("Kayakata",5,4)
```

MINIF *Basic MINIF function from excel*

Description

It acts similiarly to Excel's MINIF function. It returns the minimum value from an array after testing for certain criterias.

Usage

```
MINIF(range, criteria, min_range)
```

Arguments

range	Which range should it check the criteria against?
criteria	What should be checked?
min_range	From which range should it return the minimum from?

Value

In this case we get the minimum value of Sepal Length from Species which are virginica. Function will always return numeric class.

Examples

```
MINIF(iris$Species,"virginica",iris$Sepal.Length)
```

MINIFS *Basic MINIFS function from excel*

Description

It acts similiarly to Excel's MINIFS function. It returns the minimum value from an array after testing for several criterias.

Usage

```
MINIFS(
  min_range,
  criteria_range1,
  criteria1,
  criteria_range2 = TRUE,
  criteria2 = TRUE,
  criteria_range3 = TRUE,
  criteria3 = TRUE,
  criteria_range4 = TRUE,
  criteria4 = TRUE,
  criteria_range5 = TRUE,
  criteria5 = TRUE,
  criteria_range6 = TRUE,
  criteria6 = TRUE,
  criteria_range7 = TRUE,
  criteria7 = TRUE,
  criteria_range8 = TRUE,
  criteria8 = TRUE,
  criteria_range9 = TRUE,
  criteria9 = TRUE,
  criteria_range10 = TRUE,
  criteria10 = TRUE
)
```

Arguments

min_range	From which range should it return the minimum from?

`criteria_range1, criteria_range2, criteria_range3, criteria_range4, criteria_range5, criteria_range6, criteria_range7, criteria_range8, criteria_range9, criteria_range10`
: Which range should the criteria tested for.Only criteria_range1 is mandatory,others are optional.

`criteria1, criteria2, criteria3, criteria4, criteria5, criteria6, criteria7, criteria8, criteria9, criteria10`
: What criteria should the range be checked against. Only criteria1 is mandatory, others are optional.

Value

This example returns minimum value of Petal Length of species setosa who have petal width 0.2. Function will always return numeric class.

Examples

```
MINIFS(iris$Petal.Length,iris$Species,"setosa",iris$Petal.Width,0.2)
```

MONTH *Basic MONTH function from excel*

Description

It acts similiarly to Excel's MONTH function. It extracts the month part from the date.

Usage

```
MONTH(date)
```

Arguments

`date`	Enter the date to get the month from.

Value

In this case the function will give you 12 as it is the month of the date we have specified.Function will always return numeric class.

Examples

```
MONTH(DATE(2020,12,1))
```

NOT *Basic NOT function from excel*

Description

It acts similiarly to Excel's NOT function. It returns the opposite of the logical you specify.

Usage

```
NOT(logical)
```

Arguments

`logical`	Enter the logical to get the opposite logical of it. For example if you input TRUE, it will get FALSE.

Value

in the first example it will return false while in the 2nd example it will return true. Function will always return logical class.

Examples

```
NOT(TRUE)
NOT(FALSE)
```

NOW *Basic NOW function from excel*

Description

It acts similiarly to Excel's NOW function. It gives the system time in character format.

Usage

```
NOW()
```

Value

As the function has no arguments it simply returns current system time in character format. Function will always return character class.

Examples

```
NOW()
```

OR *Basic OR function from excel*

Description

It acts similiarly to Excel's OR function. Logical operator where if at least only one logical is true it returns true.

Usage

```
OR(
  logical1,
  logical2 = FALSE,
  logical3 = FALSE,
  logical4 = FALSE,
  logical5 = FALSE,
  logical6 = FALSE,
  logical7 = FALSE,
  logical8 = FALSE,
  logical9 = FALSE,
  logical10 = FALSE,
  logical11 = FALSE,
  logical12 = FALSE,
  logical13 = FALSE,
  logical14 = FALSE,
  logical15 = FALSE,
  logical16 = FALSE,
  logical17 = FALSE,
  logical18 = FALSE,
  logical19 = FALSE,
  logical20 = FALSE,
  logical21 = FALSE,
  logical22 = FALSE,
  logical23 = FALSE,
  logical24 = FALSE,
  logical25 = FALSE,
  logical26 = FALSE,
  logical27 = FALSE,
  logical28 = FALSE,
  logical29 = FALSE,
  logical30 = FALSE,
  logical31 = FALSE,
  logical32 = FALSE
)
```

Arguments

`logical1, logical2, logical3, logical4, logical5, logical6, logical7, logical8, logical9, logical10, logical11, logical12, logical13, logical14, logical15, logical16, logical17, logical18, logical19, logical20, logical21, logical22, logical23, logical24, logical25, logical26, logical27, logical28, logical29, logical30, logical31, logical32`

Give the function a logical argument. The one that returns either TRUE or FALSE.

Value

In this example either if species is virginica or sepal length is more than 6 then it returns true. Function will always return logical class.

Examples

```
OR(iris$Species == "virginica",iris$Sepal.Length > 6)
```

RAND *Basic RAND function from excel.*

Description

It acts similiarly to Excel's RAND function. No need to specify the arguments/parameters. It gives you the random number from 0 to 1.

Usage

```
RAND()
```

Value

This example simply returns a number from 0 to 1. Function will always return numeric class.

Examples

```
RAND()
```

RANDBETWEEN *Basic RANDBETWEEN function from excel*

Description

It acts similiarly to Excel's RANDBETWEEN function. it takes several arguments like bottom,top and number, you specify the floor, ceiling and the amount of numbers you want to generate and it gives you the random between the floor and ceiling.

Usage

```
RANDBETWEEN(bottom, top, number = 1)
```

Arguments

bottom	Give the function a bottom floor for the randbetween
top	Give the function a top ceiling for the randbetween
number	How many numbers should it generate?

Value

In the first example we get only 1 number from 1 to 100, while in the second example we get 3 numbers from 1 to 100 as the argument number is specified 3. Function will always return numeric class.

Examples

```
RANDBETWEEN(1,100, number = 1)
RANDBETWEEN(1,100, number = 3)
```

REPT *Basic REPT function from excel*

Description

It acts similiarly to Excel's REPT function. Repeat the text as many times as you want.

Usage

```
REPT(text, number_times, AsOne = TRUE)
```

Arguments

text	Which text should it repeat n time?
number_times	How many times should the function repeat the given text.
AsOne	Should function concatenate the text or should it return seperately as a vector(Vector is same as array in Excel)

Value

In the first example we repeat "Oi" 2 times and it is coerced together as one string. In the second example we don't coerce it together but it still repeats the "Oi" two times. Function will always return character class.

Examples

```
REPT("Oi",2,AsOne = TRUE)
REPT("Oi",2,AsOne = FALSE)
```

RIGHT *Basic RIGHT function from excel*

Description

It acts similiarly to Excel's RIGHT function. It takes the string and takes the amount of characters you want to extract from it.

Usage

```
RIGHT(text, num_chars)
```

Arguments

`text`	from where should it get the characters
`num_chars`	how many characters should it get?

Value

In this example we take "Kayakata" and extract 4 characters from the right handside of the string. Functions will always return character class.

Examples

```
RIGHT("Kayakata",4)
```

Sales *Random Sales Data*

Description

A dataset containing randomly generated Sales data.

Usage

```
Sales
```

Format

A data frame of 24 rows and 4 columns

Names Names of salesman

Country Countries of the salesman

Cost Cost of each salesman

Sales Amount of sales each salesman generates

Source

Randomly generated data

STDEV *Basic STDEV function from excel*

Description

It acts similiarly to Excel's STDEV function. It calculates the standard deviation from the numbers you give it.

Usage

```
STDEV(
  number1,
  number2 = NA,
  number3 = NA,
  number4 = NA,
  number5 = NA,
  number6 = NA,
  number7 = NA,
  number8 = NA,
  number9 = NA,
  number10 = NA,
```

```
  number11 = NA,
  number12 = NA,
  number13 = NA,
  number14 = NA,
  number15 = NA,
  number16 = NA,
  number17 = NA,
  number18 = NA,
  number19 = NA,
  number20 = NA,
  number21 = NA,
  number22 = NA,
  number23 = NA,
  number24 = NA
)
```

Arguments

number1, number2, number3, number4, number5, number6, number7, number8, number9, number10, number11, number12, number13, number14, number15, number16, number17, number18, number19, number20, number21, number22, number23, number24

From which numbers should the function calculate the standard deviation. Sames goes for other number arguments as well. If you want to specify several numbers simply go: STDEV(2,2,1,2). No need to put them into a vector.

Value

In this example we simply calculate standard deviation of the given numbers. Function will always return numeric class.

Examples

```
STDEV(2,1,3,1)
```

Streets *Random Salesman Streets Data*

Description

A dataset containing randomly generated Streets data.

Usage

```
Streets
```

Format

A data frame of 4 rows and 2 columns

Country Names of salesman

Street Street the salesman lives on

Source

Randomly generated data

`SUBSTITUTE` *Basic SUBSTITUTE function from excel*

Description

It acts similiarly to Excel's SUBSTITUTE function. If you want to substitute the characters by certain characters you should use this function.

Usage

```
SUBSTITUTE(text, old_text, new_text)
```

Arguments

`text`	Where should it substitute the characters
`old_text`	Which text should it substitute
`new_text`	What should it substitute with.

Value

In this example we take text "CRAN", we take the old text "RAN" and replace it with "out" which in return gives us "Cout". Function will always return character class.

Examples

```
SUBSTITUTE("CRAN","RAN","out")
```

SUMIF *Basic SUMIF function from excel*

Description

It acts similiarly to Excel's SUMIF function. It sums the values where certain criterias are met.

Usage

```
SUMIF(range, criteria, sum_range)
```

Arguments

range	Which range should it check the criteria against.
criteria	what criteria should it check in range
sum_range	Which range should it sum

Value

In this case we are summing Sepal length of species which are virginica. Function will always return numeric class.

Examples

```
SUMIF(iris$Species,"virginica",iris$Sepal.Length)
```

SUMIFS *Basic SUMIFS function from excel*

Description

It acts similiarly to Excel's SUMIFS function. It sums the values where several criterias are met(it mimics and expression for criterias).

Usage

```
SUMIFS(
  sum_range,
  criteria_range1,
  criteria1,
  criteria_range2 = TRUE,
  criteria2 = TRUE,
  criteria_range3 = TRUE,
  criteria3 = TRUE,
  criteria_range4 = TRUE,
```

```
  criteria4 = TRUE,
  criteria_range5 = TRUE,
  criteria5 = TRUE,
  criteria_range6 = TRUE,
  criteria6 = TRUE,
  criteria_range7 = TRUE,
  criteria7 = TRUE,
  criteria_range8 = TRUE,
  criteria8 = TRUE,
  criteria_range9 = TRUE,
  criteria9 = TRUE,
  criteria_range10 = TRUE,
  criteria10 = TRUE
)
```

Arguments

sum_range	Which range should it sum
criteria_range1, criteria_range2, criteria_range3, criteria_range4, criteria_range5, criteria_range6, criteria_range7, criteria_range8, criteria_range9, criteria_range10	Which range should it check the criteria against. Only criteria_range1 is mandatory, others are optional.
criteria1, criteria2, criteria3, criteria4, criteria5, criteria6, criteria7, criteria8, criteria9, criteria10	what criteria should it check in range. Only criteria1 is mandatory, others are optional.

Value

In this example we sum the petal length of all setosa species which have petal width of 0.2. Function will always return numeric class.

Examples

```
SUMIFS(iris$Petal.Length,iris$Species,"setosa",iris$Petal.Width,0.2)
```

TODAY *Basic TODAY function from excel*

Description

It acts similiarly to Excel's TODAY function.No need to give the arguments. Function returns the system date.

Usage

```
TODAY()
```

Value

Example returns the system date as the function does, nothing specific. Function will always return Date class.

Examples

```
TODAY()
```

UPPER *Basic UPPER function from excel*

Description

It acts similiarly to Excel's UPPER function. It takes the string and coverts all of it's characters to uppercase.

Usage

```
UPPER(text)
```

Arguments

text	Give this function the text to capitalize all the letters.Give this function words with a vector if you want to perform it on several texts.

Value

In this case we have specified "is this sparta?" and it has returned all the characters in uppercase as expected. Function will always return character class.

Examples

```
UPPER("is this sparta?")
```

VLOOKUP *Basic VLOOKUP function from excel*

Description

It acts similiarly to Excel's VLOOKUP function with some extra arguments. It takes the value that you want to take from another table and returns the corresponding value from another table. Basically it's an SQL Left Join.

Usage

```
VLOOKUP(
  lookup_from_table,
  lookup_column_value = "Name of the column to lookup",
  lookup_where_table,
  lookup_where_table_column = "Name of the column to compare",
  return_which_column = "Name of the column to return"
)
```

Arguments

lookup_from_table	The table it should lookup values from
lookup_column_value	which column should be looked up
lookup_where_table	which table should it look for the values in
lookup_where_table_column	Which column should it look for the values in.
return_which_column	Which column should it return

Value

In this case we have built-in database Sales and Street. We try to merge these 2 tables to see on which street are the salesman based on their countries. Function can return numeric,character,logical or any other class, it depends on what is in the table you are looking up the value in.

Examples

```
VLOOKUP(Sales,"Country",Streets,"Country","Street")
```

WEEKDAY *Basic WEEKDAY function from excel*

Description

It acts similiarly to Excel's WEEKDAY function. It tells you the weekday of the date's day either in number format or character format.

Usage

```
WEEKDAY(date, return = "number")
```

Arguments

date	What date should it take to get the weekday from. For example: "23-06-2020"
return	Should it return number or should it return the day in the characters format. Specify in quotes number if you want it to return number, specify character if you want to get the character, like Monday

Value

In the first case we get the number formatted 1st February of 2020 which is 6. We plug in the same date in the 2nd example and we specify the return argument "character", therefore it gives us the "Saturday" in character format. Function returns either character or numeric class.

Examples

```
WEEKDAY(DATE(2020,2,1),return = "number")
WEEKDAY(DATE(2020,2,1),return = "character")
```

YEAR *Basic YEAR function from excel*

Description

It acts similiarly to Excel's YEAR function. Function will extract year component of your date.

Usage

```
YEAR(date)
```

Arguments

date	Give the date argument so it can extract year from the date. Preferable to give the date via DATE function of this package.

Value

In this example function returns 2020 as it is the year part of the date specified. Function will always return numeric class..

Examples

```
YEAR(DATE(2020,1,1))
```

Index