**Sort a Vector**

**Usage**

sort(x, partial = NULL, na.last = NA)

**Details**

If partial is non NULL, it is taken to contain indexes of elements of x which are to be placed in their correct positions by partial sorting. After the sort, the values specified in partial are in their correct position in the sorted array. Any values smaller than these values are guaranteed to have a smaller index in the sorted array and any values which are greater are guaranteed to have a bigger index in the sorted array.

If na.last is NA, sort removes missing values. If na.last is TRUE, missing values are put last, if FALSE, missing values are put first.

### Examples

data(swiss)

x <- swiss$Education[1:25]

x; sort(x); sort(x, partial = c(10,15))

median # shows you another example for `partial'

**RowSums, colSums, rowMeans and ColMeans in R**

**rowSums** function in R sums up all the rows and returns the output

**colSums** function in R sums up all the columns and returns the output

**rowMeans** function in R find the mean of all the rows and returns the output

**colMeans** function in R find the mean of all the columns and returns the output

#### **rowSums() in R:**

lets use iris data set to depict example on rowSums function in R

#rowSums(iris[,5])

The above function calculates [sum](http://www.datasciencemadesimple.com/sum-function-in-r/) of all the rows of the iris data set. We will be neglecting fifth column because it is categorical. As we have 150 rows in the iris data set, the output will be with 150 elements.

So the output will be

[1] 10.2 9.5 9.4 9.4 10.2 11.4 9.7 10.1 8.9 9.6 10.8 10.0 9.3 8.5 11.2 12.0  
[17] 11.0 10.3 11.5 10.7 10.7 10.7 9.4 10.6 10.3 9.8 10.4 10.4 10.2 9.7 9.7 10.7  
[33] 10.9 11.3 9.7 9.6 10.5 10.0 8.9 10.2 10.1 8.4 9.1 10.7 11.2 9.5 10.7 9.4  
[49] 10.7 9.9 16.3 15.6 16.4 13.1 15.4 14.3 15.9 11.6 15.4 13.2 11.5 14.6 13.2 15.1  
[65] 13.4 15.6 14.6 13.6 14.4 13.1 15.7 14.2 15.2 14.8 14.9 15.4 15.8 16.4 14.9 12.8  
[81] 12.8 12.6 13.6 15.4 14.4 15.5 16.0 14.3 14.0 13.3 13.7 15.1 13.6 11.6 13.8 14.1  
[97] 14.1 14.7 11.7 13.9 18.1 15.5 18.1 16.6 17.5 19.3 13.6 18.3 16.8 19.4 16.8 16.3  
[113] 17.4 15.2 16.1 17.2 16.8 20.4 19.5 14.7 18.1 15.3 19.2 15.7 17.8 18.2 15.6 15.8  
[129] 16.9 17.6 18.2 20.1 17.0 15.7 15.7 19.1 17.7 16.8 15.6 17.5 17.8 17.4 15.5 18.2  
[145] 18.2 17.2 15.7 16.7 17.3 15.8

#### **colSums()in R:**

lets use iris data set to depict example on colSums function in R

#colSums(iris[,5])

The above function calculates [sum](http://www.datasciencemadesimple.com/sum-function-in-r/) of all the columns of the iris data set. We will be neglecting fifth column because it is categorical. As we have 4 columns in the iris data set, the output will be with 4 elements.

So the output will be

Sepal.Length      Sepal.Width     Petal.Length     Petal.Width

876.5                  458.6              563.7                179.9

#### **rowMeans() in R:**

lets use iris data set to depict example on rowMeans function in R

|  |  |
| --- | --- |
| 1  2  3 | # rowMeans function in R    rowMeans(iris[,-5]) |

The above function calculates [Mean](http://www.datasciencemadesimple.com/mean-function-in-r/) of all the rows of the iris data set. We will be neglecting fifth column because it is categorical. As we have 150 rows in the iris data set, the output will be with 150 elements.

So the output will be

[1] 2.550 2.375 2.350 2.350 2.550 2.850 2.425 2.525 2.225 2.400 2.700 2.500 2.325  
[14] 2.125 2.800 3.000 2.750 2.575 2.875 2.675 2.675 2.675 2.350 2.650 2.575 2.450  
[27] 2.600 2.600 2.550 2.425 2.425 2.675 2.725 2.825 2.425 2.400 2.625 2.500 2.225  
[40] 2.550 2.525 2.100 2.275 2.675 2.800 2.375 2.675 2.350 2.675 2.475 4.075 3.900  
[53] 4.100 3.275 3.850 3.575 3.975 2.900 3.850 3.300 2.875 3.650 3.300 3.775 3.350  
[66] 3.900 3.650 3.400 3.600 3.275 3.925 3.550 3.800 3.700 3.725 3.850 3.950 4.100  
[79] 3.725 3.200 3.200 3.150 3.400 3.850 3.600 3.875 4.000 3.575 3.500 3.325 3.425  
[92] 3.775 3.400 2.900 3.450 3.525 3.525 3.675 2.925 3.475 4.525 3.875 4.525 4.150  
[105] 4.375 4.825 3.400 4.575 4.200 4.850 4.200 4.075 4.350 3.800 4.025 4.300 4.200  
[118] 5.100 4.875 3.675 4.525 3.825 4.800 3.925 4.450 4.550 3.900 3.950 4.225 4.400  
[131] 4.550 5.025 4.250 3.925 3.925 4.775 4.425 4.200 3.900 4.375 4.450 4.350 3.875  
[144] 4.550 4.550 4.300 3.925 4.175 4.325 3.950

#### **colMeans function in R:**

lets use iris data set to depict example on colMeans function in R

|  |  |
| --- | --- |
| 1  2  3 | # colMeans function in R    colMeans(iris[,-5]) |

The above function calculates [mean](http://www.datasciencemadesimple.com/mean-function-in-r/) of all the columns of the iris data set. We will be neglecting fifth column because it is categorical. As we have 4 columns in the iris data set, the output will be with 4 elements.

So the output will be

Sepal.Length     Sepal.Width      Petal.Length      Petal.Width

5.843333            3.057333          3.758000          1.199333