Numeric , Arithmetic , Assignment and Vectors

25-12-2024

In this we introduces the use of R for basic mathematics, storing results for later use, and vectors, which are essential in R. In this we emphasizes that much of R functionality is designed with vector operations in mind.

Arithmetic functions

```
10+10   # addition
```

```
## [1] 20
```

```
10/10   # division
```

```
## [1] 1
```

```
10-10   # subtraction
```

```
## [1] 0
```

```
10*10   # multiplication
```

```
## [1] 100
```

```
sqrt(10)# square root
```

```
## [1] 3.162278
```

```
10^2    # power
```

```
## [1] 100
```

logarithm functions

```
log(10)   # log
```

```
## [1] 2.302585
```

```r
exp(10)   # exponent
```

```
## [1] 22026.47
```

E-notation

```r
1e10
```

```
## [1] 1e+10
```

```r
1e10/100  # ten raise to the power
```

```
## [1] 1e+08
```

Assignment and Variables

The assignment operator (<-) is used to store values in variables.This stores the value on the right side of the operator to the object on the left.

```r
# objectifying the values
x<-10/3   # assign a value to a x
x
```

```
## [1] 3.333333
```

```r
# Once a value is stored in a variable, it can be used in other calculations.
x+3
```

```
## [1] 6.333333
```

```r
x*8
```

```
## [1] 26.66667
```

```r
x/3
```

```
## [1] 1.111111
```

Vectors

Vectors are one-dimensional arrays that can hold multiple values of the same data type.The c() function (which stands for `concatenate`) is used to create vectors.

```r
my_vec<-c(1,2,3,4,5)
my_vec
```

```
## [1] 1 2 3 4 5
```

vector arithmetic

Arithmetic operations can be performed on vectors.When an arithmetic operation is performed on two vectors, R performs the operation element-by-element.

```
v1<-c(2,4,6,8,5)
v2<-c(3,5,7,9,11)

v1-v2
```

```
## [1] -1 -1 -1 -1 -6
```

```
v1/v2
```

```
## [1] 0.6666667 0.8000000 0.8571429 0.8888889 0.4545455
```

```
v1*v2
```

```
## [1]  6 20 42 72 55
```

```
v1+v2
```

```
## [1]  5  9 13 17 16
```

Recycling

If two vectors are of different lengths, R will recycle the shorter vector to match the length of the longer vector.

```
v1
```

```
## [1] 2 4 6 8 5
```

```
v1+1
```

```
## [1] 3 5 7 9 6
```

```
v1 + c(1,2)  # c(1,2) was recycled to c(1,2,1,2) to match the length of v1.
```

```
## Warning in v1 + c(1, 2): longer object length is not a multiple of shorter
## object length
```

```
## [1]  3  6  7 10  6
```

Regular Sequences

R provides functions for generating regular sequences of numbers.The colon operator (:) is used to create a sequence of integers.

```r
y<-1:20  # generate range of no. form 1 to 20.
y
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

```r
# The seq() function creates more complex sequences.
seq(from=2,to=10)               # no. seq range from 2 to 10
```

```
## [1]  2  3  4  5  6  7  8  9 10
```

```r
seq(from=2,to=10,by=2)          # no. seq range by gap of 2.
```

```
## [1]  2  4  6  8 10
```

```r
seq(from=2,to=10,length.out=9)  # no. seq range of length 9.
```

```
## [1]  2  3  4  5  6  7  8  9 10
```

Sub-setting Vectors

The square brackets ([ ]) are used to extract elements from a vector.The index or indices of the element(s) to be extracted are placed inside the brackets.

```r
# value extraction from vectors.
v1
```

```
## [1] 2 4 6 8 5
```

```r
v1[2]
```

```
## [1] 4
```

```r
v1[c(2,5)]
```

```
## [1] 4 5
```

```r
# To omit elements, use negative index numbers inside the square brackets.
v1[-2]
```

```
## [1] 2 6 8 5
```

```r
v1[-c(2,4)]
```

```
## [1] 2 6 5
```

```r
# You can use logical vectors (contain only TRUE or FALSE) to extract elements from vectors.
v1
```

```
## [1] 2 4 6 8 5
```

```r
v1[c(TRUE,FALSE,TRUE,TRUE,FALSE)] # false means explode it form vector.
```

```
## [1] 2 6 8
```

Vector Functions

R provides many functions for working with vectors.Some Useful Functions are as follows:

```r
v2
```

```
## [1]  3  5  7  9 11
```

```r
length(v2)
```

```
## [1] 5
```

```r
sum(v2)
```

```
## [1] 35
```

```r
prod(v2)
```

```
## [1] 10395
```

```r
sqrt(v2)
```

```
## [1] 1.732051 2.236068 2.645751 3.000000 3.316625
```

```r
sort(v2)
```

```
## [1]  3  5  7  9 11
```

Vectorized Functions meaning that they can operate on entire vectors at once.

```r
v3<-c(2.3456,3.5678,3.456,1.267)
round(v3,2) # round to two decimal places
```

```
## [1] 2.35 3.57 3.46 1.27
```

```r
log(v3)      # log of a vector that is v3.
```

```
## [1] 0.8525412 1.2719492 1.2401119 0.2366519
```