R Basics

Md Rasheduzzaman

2024-08-13

Data types, vectors, functions, R packages

Table of contents

0.1 R as Calculator

```
3+2  # Run
```

```
[1] 5
```

```
3-2
```

```
[1] 1
```

```
3*2
```

```
[1] 6
```

```
3/2
```

```
[1] 1.5
```

```
3%/%2 #integer division
```

```
[1] 1
```

```
5%%3 #modulus division
```

```
[1] 2
```

```
(10 - 5) * (2 + 4)  # Importance of brackets
```

```
[1] 30
```

```
10 - 5 * 2 + 4
```

```
[1] 4
```

```
(10 - 5)  * (2 + 4) # The use of brackets change the order
```

```
[1] 30
```

```
############################
```

```
7/(1+3)
```

```
[1] 1.75
```

```
7/1+3
```

```
[1] 10
```

```
7/(1+3); 7/1+3
```

```
[1] 1.75
```

```
[1] 10
```

```r
1+2; log(1); 1/10
```

```
[1] 3
```

```
[1] 0
```

```
[1] 0.1
```

```r
##########################################
## You can also give numbers a name,
## known as variables

a=5
a
```

```
[1] 5
```

```r
a*7
```

```
[1] 35
```

```r
a=a+10
a
```

```
[1] 15
```

```r
a*3
```

```
[1] 45
```

```r
a=a+10
b=a+10

a
```

```
[1] 25
```

```r
a^2 #power/exponent 2 or square
```

```
[1] 625
```

```r
a**2 #same as ^ (power)
```

```
[1] 625
```

```r
#Task: normal division, integer division and modulus division of 7 and 3
######################
7/3
```

```
[1] 2.333333
```

```r
7%/%3
```

```
[1] 2
```

```r
7%%3
```

```
[1] 1
```

```r
x=5/3 #normal division
x
```

```
[1] 1.666667
```

```r
y=5%/%3 #Integer division
y
```

```
[1] 1
```

```r
z=5%%3 #modulus
z
```

```
[1] 2
```

```r
x=5/3
x
```

```
[1] 1.666667
```

```r
floor(x)
```

```
[1] 1
```

```r
# -> 1 ## previous largest integer/divident
ceiling(x)
```

```
[1] 2
```

```r
# -> 2 ## next smallest integer/reminder
round(x) # -> 2 (decided based on the number)
```

```
[1] 2
```

```r
x=5/4
x
```

```
[1] 1.25
```

```r
floor(x)
```

```
[1] 1
```

```r
ceiling(x) #in the form of nearby integer
```

```
[1] 2
```

```r
x%%2
```

```
[1] 1.25
```

```
#Task;
#make a varible having a value of 15
#do different types of arithmatic operations

y=15
y
```

```
[1] 15
```

```
y+5
```

```
[1] 20
```

```
y-5
```

```
[1] 10
```

```
y*5
```

```
[1] 75
```

```
y/5
```

```
[1] 3
```

```
(y+4)*(4-y)
```

```
[1] -209
```

```
y%/%4
```

```
[1] 3
```

```
y%%4
```

```
[1] 3
```

```
floor(y/4)
```

```
[1] 3
```

```
ceiling(y/4)
```

```
[1] 4
```

```
y/4
```

```
[1] 3.75
```

```
round(y/4)
```

```
[1] 4
```

```
##########################################
## Exercise :

(2+3) + (2*3) - (6/3) -3^2
```

```
[1] 0
```

```
##########################################
######### logical operations #############
a=5
b=7

a=5
b=8

a==b #FALSE
```

```
[1] FALSE
```

```
a!=b # TRUE
```

```
[1] TRUE
```

```
a>b # FALSE and so on for the remaining
```

```
[1] FALSE
```

```
a<b
```

```
[1] TRUE
```

```
a>=b
```

```
[1] FALSE
```

```
a<=b
```

```
[1] TRUE
```

```
a<b | a>b
```

```
[1] TRUE
```

```
a<b & a>b
```

```
[1] FALSE
```

```
a<b | a>=b # ?
```

```
[1] TRUE
```

```
a<b & a>=b # ?
```

```
[1] FALSE
```

```
# | (pipe) ==>

#######################################
### help ############################
?log
help(log)
example(log)
```

```
log> log(exp(3))
[1] 3

log> log10(1e7) # = 7
[1] 7

log> x <- 10^-(1+2*1:9)

log> cbind(deparse.level=2, # to get nice column names
log+       x, log(1+x), log1p(x), exp(x)-1, expm1(x))
          x   log(1 + x)     log1p(x)   exp(x) - 1     expm1(x)
 [1,] 1e-03 9.995003e-04 9.995003e-04 1.000500e-03 1.000500e-03
 [2,] 1e-05 9.999950e-06 9.999950e-06 1.000005e-05 1.000005e-05
 [3,] 1e-07 1.000000e-07 1.000000e-07 1.000000e-07 1.000000e-07
 [4,] 1e-09 1.000000e-09 1.000000e-09 1.000000e-09 1.000000e-09
 [5,] 1e-11 1.000000e-11 1.000000e-11 1.000000e-11 1.000000e-11
 [6,] 1e-13 9.992007e-14 1.000000e-13 9.992007e-14 1.000000e-13
 [7,] 1e-15 1.110223e-15 1.000000e-15 1.110223e-15 1.000000e-15
 [8,] 1e-17 0.000000e+00 1.000000e-17 0.000000e+00 1.000000e-17
 [9,] 1e-19 0.000000e+00 1.000000e-19 0.000000e+00 1.000000e-19
```

```
?sd

############################
```

```
# create your first vector

x=c(1,2,3,4,5)
y=c(3,6,9,12,15)
y
```

```
[1]  3  6  9 12 15
```

```
x
```

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

```
length(x)
```

```
[1] 5
```

```
length(y)
```

```
[1] 5
```

```
mode(x)
```

```
[1] "numeric"
```

```
is(x)
```

```
[1] "numeric" "vector"
```

```
mode(y)
```

```
[1] "numeric"
```

```
is(y)
```

```
[1] "numeric" "vector"
```

```
x= c(1, 2, 3, 4, 5, 6, 7, 8, 9) # c="concatenate"
# x is a vector
x
```

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

```r
mode(x)    # mode of x
```

```
[1] "numeric"
```

```r
is(x)     # type of x
```

```
[1] "numeric" "vector"
```

```r
length(x) #length of x
```

```
[1] 9
```

```r
DNA=c("A", "T", "G", "C")
DNA
```

```
[1] "A" "T" "G" "C"
```

```r
mode(DNA)
```

```
[1] "character"
```

```r
is(DNA)
```

```
[1] "character"           "vector"              "data.frameRowLabels"
[4] "SuperClassMethod"
```

```r
length(DNA)
```

```
[1] 4
```

```r
#Task: character vector of 1, 3, 5, 7


char=c("1", "3", "5", "7")
char
```

```
[1] "1" "3" "5" "7"
```

```
DNA2=c("ATGTGTCA", "GTCA", "GTCATC")
#Task: mode, is function, length

mode(DNA2)
```

```
[1] "character"
```

```
is(DNA2)
```

```
[1] "character"           "vector"              "data.frameRowLabels"
[4] "SuperClassMethod"
```

```
length(DNA2)
```

```
[1] 3
```

```
logicals=c(TRUE, TRUE, FALSE, TRUE, FALSE)
logicals
```

```
[1]  TRUE  TRUE FALSE  TRUE FALSE
```

```
mode(logicals)
```

```
[1] "logical"
```

```
length(logicals)
```

```
[1] 5
```

```
is(logicals)
```

```
[1] "logical" "vector"
```

```r
y= c("1", "2", "3", "4")

mode(y)
```

```
[1] "character"
```

```r
dec=c(10,20,30,60,80,90,100,50,40)
dec
```

```
[1]  10  20  30  60  80  90 100  50  40
```

```r
dec[2]
```

```
[1] 20
```

```r
dec[7]
```

```
[1] 100
```

```r
dec[3]
```

```
[1] 30
```

```r
DNA2=c("ATGTGTCA", "GTCA", "GTCATC")
DNA2[2]
```

```
[1] "GTCA"
```

```r
DNA2[3]
```

```
[1] "GTCATC"
```

```r
yz=c(3, 4, 7, 9, 10, 5)
yz
```

```
[1]  3  4  7  9 10  5
```

```r
#Task: 3rd, 4th, and 2nd element from yz

yz[3]
```

```
[1] 7
```

```r
yz[4]
```

```
[1] 9
```

```r
yz[2]
```

```
[1] 4
```

```r
s= c("AATTGCCC", "ATGCATT", "AACCGTTG")
s[1]
```

```
[1] "AATTGCCC"
```

```r
# Task: find the others with indexing
s[2]
```

```
[1] "ATGCATT"
```

```r
s[3]
```

```
[1] "AACCGTTG"
```

```r
##########################################
########## vector operations ############
# Most standard mathematical functions work with vectors.
x= c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

x+x  #goes index-wise
```

```
 [1]  2  4  6  8 10 12 14 16 18 20
```

```r
mode(x)
```

```
[1] "numeric"
```

```r
is(x)
```

```
[1] "numeric" "vector"
```

```r
y= c(5, 10 ,15, 20, 25, 30, 35, 40, 45, 50)
y/x
```

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

```r
y * x
```

```
 [1]   5  20  45  80 125 180 245 320 405 500
```

```r
log2(x)
```

```
 [1] 0.000000 1.000000 1.584963 2.000000 2.321928 2.584963 2.807355 3.000000
 [9] 3.169925 3.321928
```

```r
#Round the values
x= c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
log2(x)
```

```
 [1] 0.000000 1.000000 1.584963 2.000000 2.321928 2.584963 2.807355 3.000000
 [9] 3.169925 3.321928
```

```r
round(log2(x))
```

```
 [1] 0 1 2 2 2 3 3 3 3 3
```

```r
####Task: Round the value for 1, 3 digits after decimal
round(log2(x), 1)
```

```
 [1] 0.0 1.0 1.6 2.0 2.3 2.6 2.8 3.0 3.2 3.3
```

```
round(log2(x), 3)
```

```
 [1] 0.000 1.000 1.585 2.000 2.322 2.585 2.807 3.000 3.170 3.322
```

```
########### Exercise 1
# Compute the difference between 2020
# and the year you started at the university and
# divide this by the difference between 2020 and the year you were born.
# Multiply this with 100 to get the percentage of your life
# you have spent at the university.
```

0.2 Basic plotting

```
a <- 1:10
b <-10:1
plot(a, b)
```