R Notebook

Basic operation in R

Pavan Pyla 22232

Date 03-02-2023

```
# Creating a variable
x <- 5
y <- 10

# Basic arithmetic operations
z <- x + y # Addition
print(z)
```

```
## [1] 15
```

```
z <- x - y # Subtraction
print(z)
```

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

```
z <- x * y # Multiplication
print(z)
```

```
## [1] 50
```

```
z <- x / y # Division

# Printing the result
print(z)
```

```
## [1] 0.5
```

```
# Working with vectors
vector1 <- c(1, 2, 3, 4, 5) # Creating a vector
vector2 <- c(5, 4, 3, 2, 1)

# Vector operations
sum_vector <- vector1 + vector2 # Addition of vectors
print(sum_vector)
```

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

```
x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
print(1/x)                          #Printing the reciprocals of each number
```

```
## [1] 0.09615385 0.17857143 0.32258065 0.15625000 0.04608295
```

```
                                    # in the declared vector
y <- c(x)
v <- 2*x + y + 1
```

Different Assignmets for vectors,variables and lists

```
apple <- c('red','green','yellow','Black')

print(apple)
```

```
## [1] "red"    "green"  "yellow" "Black"
```

```
print(class(apple))
```

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

```
d <- c(7,76,8)

print(d)
```

```
## [1]  7 76  8
```

```
print(class(d))
```

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

```
k<-x + d
```

```
## Warning in x + d: longer object length is not a multiple of shorter object
## length
```

```
print(k)
```

```
## [1] 17.4 81.6 11.1 13.4 97.7
```

```
l1<- list(c(37,4,56,3),21,sin)            #taking the values into list
print(l1)
```

```
## [[1]]
## [1] 37  4 56  3
##
## [[2]]
## [1] 21
##
## [[3]]
## function (x)  .Primitive("sin")
```

```
print(class(l1))             #printing its class and its values
```

```
## [1] "list"
```

```
for (x in 1:10) {
  print(x)                      #Loops in R
}
```

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

```
var...1 <- 6
print(var...1)
```

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

```
......hi <- 7           #Defining the variables in different ways
                        #Ways of naming variables
print(......hi)
```

```
## [1] 7
```

```
..5var <- 65
print(..5var)
```

```
## [1] 65
```

```
#matrix <- (
  #marks <- c(10.4, 5.6, 3.1, 6.4, 21.7)
  #names <- c(10.4, 5.6, 3.1, 6.4, 21.7)
#)
#print(matrix)
```

Vectors in List

```
vector1 <- c(1, 2, 3, 4, 5) # Creating a vector
vector2 <- c(5, 4, 3, 2, 1)
mylist <- c(vector1,vector2,90,'red')

print(mylist)            # Putting vectors into a list
```

```
##  [1] "1"   "2"   "3"   "4"   "5"   "5"   "4"   "3"   "2"   "1"   "90"  "red"
```

Printing the Vectors in various ways

```
vec <- c(1, 2, 3, 4, 5) # Creating a vector

1:3
```

```
## [1] 1 2 3
```

```
print(vec[1:3])     #Printing the first 3 elements in vector
```

```
## [1] 1 2 3
```

```
print(vec[-1])       #Printing the vecto in Reverse manner
```

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

```
print(vec[c(F,F,F,T,F)])   #printing the specific values of vector Using Boolen vector
```

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

Creating a Data Frame

```
df <- data.frame(a = c(1,2,3,4),b = c('apple','Orange','mango','banana'), c = c('red','yello
w','green','blue'))


print(df)
```

```
##   a      b      c
## 1 1  apple    red
## 2 2 Orange yellow
## 3 3  mango  green
## 4 4 banana   blue
```

Creating the Data Frame using existing vectors

```
df1 <- data.frame(vector1,vector2)

print(df1)
```

```
##   vector1 vector2
## 1       1       5
## 2       2       4
## 3       3       3
## 4       4       2
## 5       5       1
```

printing the Data Frame

```
print(df[1])       #printing the first column of df
```

```
##   a
## 1 1
## 2 2
## 3 3
## 4 4
```

```
print(df1[1])      #printing the first column of df1
```

```
##   vector1
## 1       1
## 2       2
## 3       3
## 4       4
## 5       5
```

```
print(df[3])       #printing the third column of df
```

```
##        c
## 1    red
## 2 yellow
## 3  green
## 4   blue
```

```
print(df[[1]][1])     #printing a specific value of data frame
```

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

Reciprocal of vector

```
vect <- c(1,2,3,4,5,6,7,8) #creating a vector

reci <- 1/vect           #creating a other vector of reciprocals of each value of the existin
g vector
print(reci)
```

```
## [1] 1.0000000 0.5000000 0.3333333 0.2500000 0.2000000 0.1666667 0.1428571
## [8] 0.1250000
```