R Programming

Assignment Operations

In [1]:

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

1 2 3 4 5

In [2]:

```
x <- c(1 : 7)
x
```

1 2 3 4 5 6 7

In [3]:

```
x <- 1 : 4
x
```

1 2 3 4

In [4]:

```
x = 12
x
```

12

Finding Datatype

In [5]:

```
typeof(1 : 4)
typeof(c(1 : 7))
```

'integer'

'integer'

Numerical Operations

In [6]:

```
log(exp(1))
```

1

In [7]:

```
log(1000, 10)
```

3

In [8]:

```
sin(pi / 3) ^ 2 + cos(pi / 3) ^ 2
```

1

In [9]:

```
log2(32)
```

5

In [10]:

```
sqrt(2)
```

1.4142135623731

In [11]:

```
x <- c(1, 2, 3, NA)
x + 3
```

4 5 6 NA

In [12]:

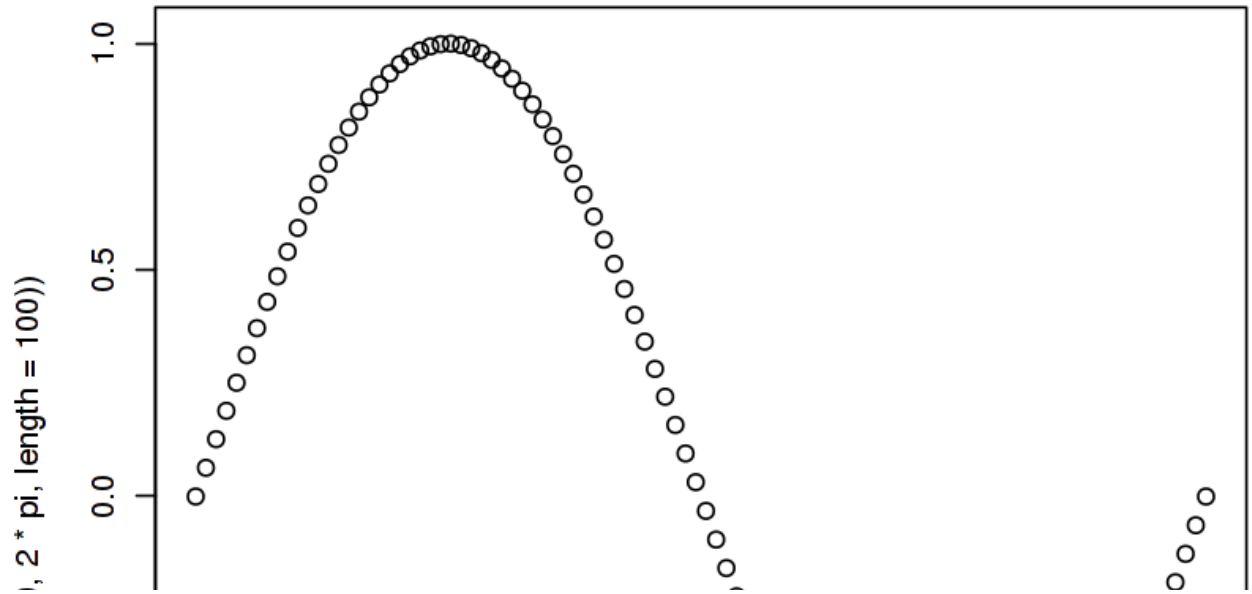
```
log(c(0, 1, 2))
```

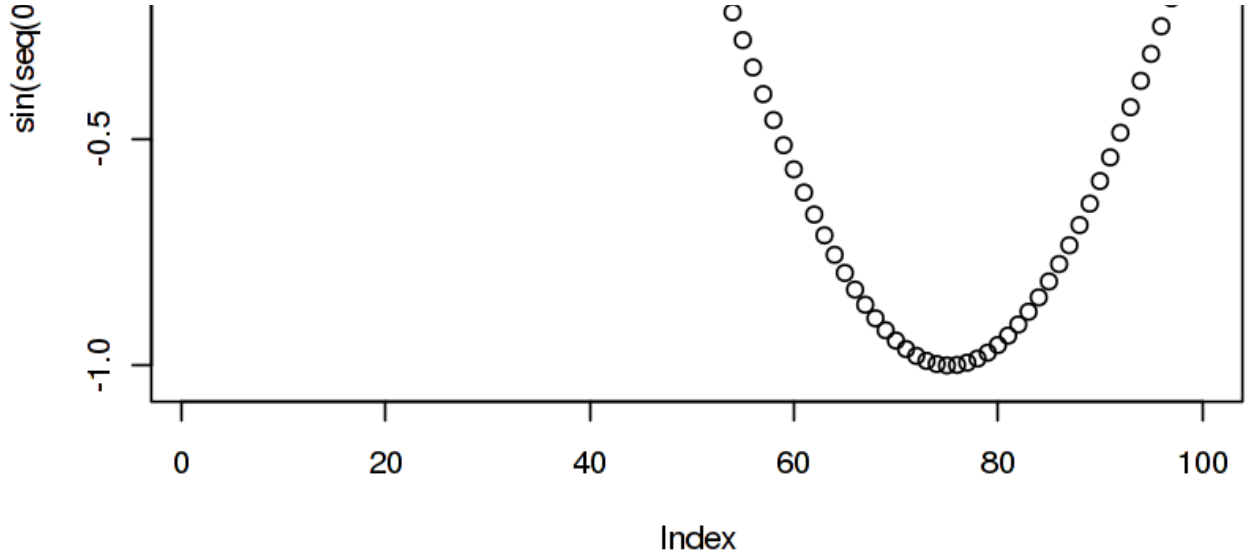
-Inf 0 0.693147180559945

Plotting Examples

In [13]:

```
plot(sin(seq(0, 2 * pi, length = 100)))
```





Data Types

- Vectors
- Lists
- Matrices
- Arrays
- Factors
- Dataframes

Basic (atomic) Data types

In [14]:

```
## Boolean
x <- T; y <- F
x; y

## Character
a <- "1"; b = 1
a; b

## Numerical
a = 5; b <- sqrt(2)
a; b
```

TRUE

FALSE

'1'

1

5

1.4142135623731

Vector Functions

```
length(object)                  no of elements or components
str(object)                     structure of an object
class(object)                   class or type of an object
names(object)                   names
c(object1, object2, ....)       combine all objects into a vector
cbind(obj1, obj2, ....)         combine all objects into a column
rbind(obj1, obj2, ....)         combine all objects into a row
ls()                            list all objects
rm(object)                      delete an object
newobject <- edit(object)       edit copy and save at newobject
fix(object)                     edit in place
```

Experiment 2

Q1. Initialize some variables in R and analyze and display them.

Q2. Initialize some variables as well as all the created variables, then delete single as well as created variable.

Q3. Inintialize roll, name, branch and display all details

Q4. Initialize variables and then sum them.

Q5. Enter 2 numbers and perform 4 arithmatic operations on them

Q6. Enter 3 digit number and find out sum of digits

Q7. Enter radius of a circle and calculate area and circumference

Q8. Enter P, T, R and calculate compound interest

Q9. Enter 2 numbers and swap them without third variable

Q10. Enter 2 numbers and implement all relational operations.

Solutions

In [15]:

```
# Q1
a = 10
b = a + 20
name = "Soumik"
first_char = "S"
is_it = F
list1 = c(1, 2, 3, 4, 5)
a; b; name; first_char; is_it; list1
```

10

30

'Soumik'

'S'

FALSE

1 2 3 4 5

In [16]:

```
# Q2
more_a = 100
more_b = "KIIT"
remove(more_a)
remove(b)
```

In [17]:

```
# Q3
roll = 1605235; name = "Soumik Rakshit"; branch = "CSE"
cat("Roll:", roll, "\n")
cat("Name:", name, "\n")
cat("Branch:", branch, "\n")
```

```
Roll: 1605235
Name: Soumik Rakshit
Branch: CSE
```

In [18]:

```
# Q4
a = 10; b = 20
c = c(100, 200); d = c(300, 400)
cat(a, " + ", b, " = ", a + b, "\n")
cat("(", c, ") + (", d, ") = (", c + d, ")\n")
```

```
10  +  20  =  30
( 100 200 ) + ( 300 400 ) = ( 400 600 )
```

In [19]:

```
# Q5
num1 = 20
num2 = 10
cat(num1, " + ", num2, " = ", num1 + num2, "\n")
cat(num1, " - ", num2, " = ", num1 - num2, "\n")
cat(num1, " * ", num2, " = ", num1 * num2, "\n")
cat(num1, " / ", num2, " = ", num1 / num2, "\n")
```

```
20  +  10  =  30
20  -  10  =  10
20  *  10  =  200
20  /  10  =  2
```

In [20]:

```
# Q6
num = 123
if(nchar(num) != 3) {
    print("Number is not of 3 digits")
} else {
    num = as.numeric(num)
    a = num %% 10
    num = num %/% 10
    b = num %% 10
    num = num %/% 10
    c = num %% 10
    num = num %/% 10
    cat("Sum: ", a + b + c)
}
```

```
Sum:  6
```

In [21]:

```
# Q7
r = 10
area = pi * r * r
circumference = 2 * pi * r
```

```
cat("Area: ", area, "\n")
cat("Circumference: ", circumference, "\n")
```

```
Area:  314.1593
Circumference:  62.83185
```

In [22]:

```
# Q8
p = 1000
n = 3
r = 10
ci = p * ((1 + (r / 100)) ** n)
cat("Compound Interest: ", ci, "\n")
```

```
Compound Interest:  1331
```

In [23]:

```
# Q9
a = 10
b = 20
a = a + b
b = a - b
a = a - b
cat("a = ", a, ", b = ", b)
```

```
a =  20 , b =  10
```

In [24]:

```
# Q10
a = 10
b = 20
a < b; a <= b; a > b; a >= b; a == b; a != b
```

TRUE

TRUE

FALSE

FALSE

FALSE

TRUE