In [14]:

```
# This Python 3 environment comes with many helpful analytics libraries installed # It is defined by the kaggle/python
docker image: https://github.com/kaggle/docker-pytho n# For example, here's several helpful packages to load in

import numpy as np # linear algebra import pandas as pd # data processing, CSV file I/O (e.g.
pd.read_csv)

# Input data files are available in the "../input/" directory. # For example, running this (by clicking run or pressing
Shift+Enter) will list all files under the input directory

        import os for for dirname, _, filenames inin os.walk('/kaggle/input'):
    for for filename inin filenames:
        print(os.path.join(dirname, filename))

# Any results you write to the current directory are saved as output.
```

```
Error in parse(text = x, srcfile = src): <text>:5:8: unexpected symbol 4: 5: import numpy
                        ^
Traceback:
```

In [ ]:

```
2+3
```

In [ ]:

```
6/3
```

In [ ]:

```
(3*8)/(2*3)
```

In [ ]:

```
log(12)
```

In [15]:

```
sqrt(121)
```

11In [16]:

```
x <- 1:10 cat(x) cat('Mean of x is',mean(x)) cat('Median
of x is',median(x)) cat('Variance of x is',var(x))
```

```
cat('Standard Deviation of x is',sd(x))
```

1 2 3 4 5 6 7 8 9 10Mean of x is 5.5Median of x is 5.5Variance of x is 9.166667Standard D eviation of x is 3.02765

In [17]:

```
cat(x,'\n') cat('Mean of x is',mean(x),'\n') cat('Median of x
is',median(x),'\n') cat('Variance of x is',var(x),'\n') cat('Standard
Deviation of x is',sd(x),'\n')
```

In 1 2 3 4 5 6 7 8 9 10 Mean of x is 5.5 Median of x is 5.5 Variance of x is 9.166667 Standard Deviation of x is 3.02765

In [18]:

```
print(x) print(paste('Mean of x is',mean(x))) print(paste('Median of x is',median(x))) print(paste('Variance of
x is', var(x))) print(paste('Standard Deviation of x is',sd(x)))
```

[1] 1 2 3 4 5 6 7 8 9 10 [1] "Mean of x is 5.5" [1] "Median of x is 5.5" [1] "Variance of x is 9.16666666666667" [1] "Standard Deviation of x is 3.02765035409749"

In [19]:

```
x <- 8+7 x
```

15

In [20]:

```
# Creation of character variable x <- "dataset" xclass(x) typeof(x) length(x) attributes(x)
```

'dataset'
'character'
'character'

1NULL

In [21]:

```
# Creation of numerical variable x <- 23 xclass(x) typeof(x) length(x) attributes(x)
```

23
'numeric'
'double'

1NULL

[22]:

```
# Creation of integer variable x <- 23L x
```

23

'integer'

'integer'

1NULL

In [23]:

```
# Creation of complex variable x <- 2 + 3i
x class(x) typeof(x) length(x) attributes(x)
```

2+3i

'complex'

'complex'

1 NULL

In [24]:

```
# Creation of logical variable x <- TRUE
TRUE x class(x) typeof(x) length(x)
attributes(x)
```

TRUE

'logical'

'logical'

1 NULL

In [25]:

```
vector()
```

In [26]:

```
vector("character", length = 5)
```

" " " " "

In [27]:

```
character(5)
```

" " " " "

In [28]:

```
class(x) typeof(x)
```

length(x) attributes(x)
In [28]:

numeric(5)

0 0 0 0 0

In [29]:

logical(5)

FALSE FALSE FALSE FALSE FALSE

In [31]:

# numeric vector having 5 elements x <-
c(12,23,34,45,56)

In [32]:

xclass(x)

12 23 34 45 56

'numeric'

In [35]:

integer vecor having 5 elements x <- c(12L,

23L ,34L, 45L, 56L) xclass(x) length(x)

12 23 34 45 56

'integer'

5In [37]:

logical vecor having 5 elements y <- c(TRUE TRUE,
TRUE, TRUE FALSE, FALSE FALSE, FALSE TRUE)

TRUE yclass(y) length(y)

TRUE TRUE FALSE FALSE TRUE

'logical'

5In [38]:

Character vecor having 3 elements z <-

c("Sarah", "Tracy", "Jon") $z_{class(z)}$ length(z)

'Sarah' 'Tracy' 'Jon'

'character'

3

In [39]:

[1] "Elements in z before "

'Sarah' 'Tracy' 'Jon'

[1] "Elements in z after adding new element"

'Sarah' 'Tracy' 'Jon' 'Annette'

In [40]:

add element at the front print("Elements in z before") z_{z} <-

c("Grey", z) print("Elements in z after adding new element") z

[1] "Elements in z before"

'Sarah' 'Tracy' 'Jon' 'Annette'

[1] "Elements in z after adding new element"

'Grey' 'Sarah' 'Tracy' 'Jon' 'Annette'

In [41]:

a <- 1:10 a

1 2 3 4 5 6 7 8 9 10

In [42]:

```
seq(10)
```

1 2 3 4 5 6 7 8 9 10

In [43]:

```
seq(from =1, to = 10, by = 0.1)
```

1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 4 4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.8 4.9 5 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8 5.9 6 6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9 7 7.1 7.2 7.3 7.4 7.5 7.6 7.7 7.8 7.9 8 8.1 8.2 8.3 8.4 8.5 8.6 8.7 8.8 8.9 9 9.1 9.2 9.3 9.4 9.5 9.6 9.7 9.8 9.9 10

In [44]:

```
1/0
```

InfIn [45]:

```
0/0
```

NaN

In [46]:

```
exp(Inf Inf)
# add element at the end print("Elements in z before ") z_z <- c(z,
"Annette") print("Elements in z after adding new element") z
```

Inf

In [47]:

```
exp(-Inf Inf)
```

0In [48]:

```
Inf-Inf Inf Inf
```

NaN

In [49]:

log(0)

-Inf

In [50]:

log(1)

0In [51]:

a <- c(1.7,"a") a

'1.7' 'a'

In [53]:

b <- c(TRUE TRUE, 2, 3, FALSE)
FALSE b

1 2 3 0

In [54]:

c <- c("a",TRUE TRUE) c

'a' 'TRUE'

In [55]:

as.numeric("1")

1In [56]:

as.numeric(c("45","66"))

45 66

In [57]:

as.numeric(c("45","66","six"))

Warning message in eval(expr, envir, enclos): “NAs introduced by coercion”

45 66 <NA>

```
exp(Inf Inf)
```

In [58]:

```
as.character(22)
```

'22'

In [59]:

```
as.character(c(22,34,56))
```

'22' '34' '56'

In [60]:

```
# Example-1 m <- matrix(nrow = 2, ncol = 2)
m
```

A matrix: 2 × 2 of type lgl

NA	**NA**
NA	**NA**

In [61]:

```
dim(m)
```

2 2

In [62]:

```
class(m)
```

'matrix'

In [63]:

```
typeof(m)
```

'logical'

In [64]:

```
# Example-2 m <- matrix(c(1:3)) print('Given Matrix is') print(m)
print('Dimensions (size) of given Matrix is') print(dim(m))
print('Class of given Matrix is') print(class(m)) print('Type of given
Matrix is') print(typeof(m))
```

```
[1] "Given Matrix is"
                    [,1] [1,] 1 [2,] 2 [3,] 3 [1] "Dimensions (size)
of given Matrix is" [1] 3 1 [1] "Class of given Matrix is" [1]
"matrix" [1] "Type of given Matrix is" [1] "integer"
```

In [66]:

```
m <- 1:10 print(m) print(dim(m)) dim(m) <- c(2, 5) print(m) print(dim(m))
```

In [1] 1 2 3 4 5 6 7 8 9 10 NULL

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

In [69]:

```
my_name <- readline(prompt="Enter name: ") my_age <- readline(prompt="Enter age: ") my_age <-
as.integer(my_age) print(paste("Hi,", my_name, "next year you will be",my_age+1, "years old."))
Enter name: Ramesh Enter age: 34 [1] "Hi, Ramesh next year you will be 35 years old."
```

[70]:

```
w <- readline() w Hello
```

'Hello'